Lecture Notes in Computer Science 5302

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

David Forsyth Philip Torr
Andrew Zisserman (Eds.)

Computer Vision – ECCV 2008

10th European Conference on Computer Vision
Marseille, France, October 12-18, 2008
Proceedings, Part I

Springer

Volume Editors

David Forsyth
University of Illinois at Urbana-Champaign, Computer Science Department
3310 Siebel Hall, Urbana, IL 61801, USA
E-mail: daf@cs.uiuc.edu

Philip Torr
Oxford Brookes University, Department of Computing
Wheatley, Oxford OX33 1HX, UK
E-mail: philiptorr@brookes.ac.uk

Andrew Zisserman
University of Oxford, Department of Engineering Science
Parks Road, Oxford OX1 3PJ, UK
E-mail: az@robots.ox.ac.uk

Library of Congress Control Number: 2008936989

CR Subject Classification (1998): I.4, I.2.10, I.5.4, I.5, I.7.5

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

ISSN 0302-9743
ISBN-10 3-540-88681-8 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-88681-5 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12538041 06/3180 5 4 3 2 1 0

Preface

Welcome to the 2008 European Conference on Computer Vision. These proceedings are the result of a great deal of hard work by many people. To produce them, a total of 871 papers were reviewed. Forty were selected for oral presentation and 203 were selected for poster presentation, yielding acceptance rates of 4.6% for oral, 23.3% for poster, and 27.9% in total.

We applied three principles. First, since we had a strong group of Area Chairs, the final decisions to accept or reject a paper rested with the Area Chair, who would be informed by reviews and could act only in consensus with another Area Chair. Second, we felt that authors were entitled to a summary that explained how the Area Chair reached a decision for a paper. Third, we were very careful to avoid conflicts of interest.

Each paper was assigned to an Area Chair by the Program Chairs, and each Area Chair received a pool of about 25 papers. The Area Chairs then identified and ranked appropriate reviewers for each paper in their pool, and a constrained optimization allocated three reviewers to each paper. We are very proud that every paper received at least three reviews.

At this point, authors were able to respond to reviews. The Area Chairs then needed to reach a decision. We used a series of procedures to ensure careful review and to avoid conflicts of interest. Program Chairs did not submit papers. The Area Chairs were divided into three groups so that no Area Chair in the group was in conflict with any paper assigned to any Area Chair in the group. Each Area Chair had a "buddy" in their group. Before the Area Chairs met, they read papers and reviews, contacted reviewers to get reactions to submissions and occasionally asked for improved or additional reviews, and prepared a rough summary statement for each of the papers in their pool.

At the Area Chair meeting, groups met separately so that Area Chairs could reach a consensus with their buddies, and make initial oral/poster decisions. We met jointly so that we could review the rough program, and made final oral/poster decisions in groups. In the separate meetings, there were no conflicts. In the joint meeting, any Area Chairs with conflicts left the room when relevant papers were discussed. Decisions were published on the last day of the Area Chair meeting.

There are three more somber topics to report. First, the Program Chairs had to deal with several double submissions. Referees or Area Chairs identified potential double submissions, we checked to see if these papers met the criteria published in the call for papers, and if they did, we rejected the papers and did not make reviews available. Second, two submissions to ECCV 2008 contained open plagiarism of published works. We will pass details of these attempts to journal editors and conference chairs to make further plagiarism by the responsible parties more difficult. Third, by analysis of server logs we discovered that

there had been a successful attempt to download all submissions shortly after the deadline. We warned all authors that this had happened to ward off dangers to intellectual property rights, and to minimize the chances that an attempt at plagiarism would be successful. We were able to identify the responsible party, discussed this matter with their institutional management, and believe we resolved the issue as well as we could have. Still, it is important to be aware that no security or software system is completely safe, and papers can leak from conference submission.

We felt the review process worked well, and recommend it to the community. The process would not have worked without the efforts of many people. We thank Lyndsey Pickup, who managed the software system, author queries, Area Chair queries and general correspondence (most people associated with the conference will have exchanged e-mails with her at some point). We thank Simon Baker, Ramin Zabih and especially Jiří Matas for their wise advice on how to organize and run these meetings; the process we have described is largely their model from CVPR 2007. We thank Jiří Matas and Dan Večerka, for extensive help with, and support of, the software system. We thank C. J. Taylor for the 3-from-5 optimization code. We thank the reviewers for their hard work. We thank the Area Chairs for their very hard work, and for the time and attention each gave to reading papers, reviews and summaries, and writing summaries.

We thank the Organization Chairs Peter Sturm and Edmond Boyer, and the General Chair, Jean Ponce, for their help and support and their sharing of the load. Finally, we thank Nathalie Abiola, Nasser Bacha, Jacques Beigbeder, Jerome Bertsch, Joëlle Isnard and Ludovic Ricardou of ENS for administrative support during the Area Chair meeting, and Danièle Herzog and Laetitia Libralato of INRIA Rhône-Alpes for administrative support after the meeting.

August 2008

Andrew Zisserman
David Forsyth
Philip Torr

Organization

Conference Chair

Jean Ponce	Ecole Normale Supérieure, France

Honorary Chair

Jan Koenderink	EEMCS, Delft University of Technology, The Netherlands

Program Chairs

David Forsyth	University of Illinois, USA
Philip Torr	Oxford Brookes University, UK
Andrew Zisserman	University of Oxford, UK

Organization Chairs

Edmond Boyer	LJK/UJF/INRIA Grenoble–Rhône-Alpes, France
Peter Sturm	INRIA Grenoble–Rhône-Alpes, France

Specialized Chairs

Frédéric Jurie	Workshops	Université de Caen, France
Frédéric Devernay	Demos	INRIA Grenoble–Rhône-Alpes, France
Edmond Boyer	Video Proc.	LJK/UJF/INRIA Grenoble–Rhône-Alpes, France
James Crowley	Video Proc.	INPG, France
Nikos Paragios	Tutorials	Ecole Centrale, France
Emmanuel Prados	Tutorials	INRIA Grenoble–Rhône-Alpes, France
Christophe Garcia	Industrial Liaison	France Telecom Research, France
Théo Papadopoulo	Industrial Liaison	INRIA Sophia, France
Jiří Matas	Conference Software	CTU Prague, Czech Republic
Dan Večerka	Conference Software	CTU Prague, Czech Republic

Program Chair Support

Lyndsey Pickup	University of Oxford, UK

Administration

Danile Herzog	INRIA Grenoble–Rhône-Alpes, France
Laetitia Libralato	INRIA Grenoble–Rhône-Alpes, France

Conference Website

Elisabeth Beaujard	INRIA Grenoble–Rhône-Alpes, France
Amaël Delaunoy	INRIA Grenoble–Rhône-Alpes, France
Mauricio Diaz	INRIA Grenoble–Rhône-Alpes, France
Benjamin Petit	INRIA Grenoble–Rhône-Alpes, France

Printed Materials

Ingrid Mattioni	INRIA Grenoble–Rhône-Alpes, France
Vanessa Peregrin	INRIA Grenoble–Rhône-Alpes, France
Isabelle Rey	INRIA Grenoble–Rhône-Alpes, France

Area Chairs

Horst Bischof	Graz University of Technology, Austria
Michael Black	Brown University, USA
Andrew Blake	Microsoft Research Cambridge, UK
Stefan Carlsson	NADA/KTH, Sweden
Tim Cootes	University of Manchester, UK
Alyosha Efros	CMU, USA
Jan-Olof Eklund	KTH, Sweden
Mark Everingham	University of Leeds, UK
Pedro Felzenszwalb	University of Chicago, USA
Richard Hartley	Australian National University, Australia
Martial Hebert	CMU, USA
Aaron Hertzmann	University of Toronto, Canada
Dan Huttenlocher	Cornell University, USA
Michael Isard	Microsoft Research Silicon Valley, USA
Aleš Leonardis	University of Ljubljana, Slovenia
David Lowe	University of British Columbia, Canada
Jiří Matas	CTU Prague, Czech Republic
Joe Mundy	Brown University, USA
David Nistér	Microsoft Live Labs/Microsoft Research, USA
Tomáš Pajdla	CTU Prague, Czech Republic
Patrick Pérez	IRISA/INRIA Rennes, France
Marc Pollefeys	ETH Zürich, Switzerland
Ian Reid	University of Oxford, UK
Cordelia Schmid	INRIA Grenoble–Rhône-Alpes, France
Bernt Schiele	Darmstadt University of Technology, Germany
Christoph Schnörr	University of Mannheim, Germany
Steve Seitz	University of Washington, USA

Richard Szeliski	Microsoft Research, USA
Antonio Torralba	MIT, USA
Bill Triggs	CNRS/Laboratoire Jean Kuntzmann, France
Tinne Tuytelaars	Katholieke Universiteit Leuven, Belgium
Luc Van Gool	Katholieke Universiteit Leuven, Belgium
Yair Weiss	The Hebrew University of Jerusalem, Israel
Chris Williams	University of Edinburgh, UK
Ramin Zabih	Cornell University, USA

Conference Board

Horst Bischof	Graz University of Technology, Austria
Hans Burkhardt	University of Freiburg, Germany
Bernard Buxton	University College London, UK
Roberto Cipolla	University of Cambridge,UK
Jan-Olof Eklundh	Royal Institute of Technology, Sweden
Olivier Faugeras	INRIA, Sophia Antipolis, France
Anders Heyden	Lund University, Sweden
Aleš Leonardis	University of Ljubljana, Slovenia
Bernd Neumann	University of Hamburg, Germany
Mads Nielsen	IT University of Copenhagen, Denmark
Tomáš Pajdla	CTU Prague, Czech Republic
Giulio Sandini	University of Genoa, Italy
David Vernon	Trinity College, Ireland

Program Committee

Sameer Agarwal	Tamara Berg	Thomas Brox
Aseem Agarwala	James Bergen	Andrés Bruhn
Jörgen Ahlberg	Marcelo Bertalmio	Antoni Buades
Narendra Ahuja	Bir Bhanu	Joachim Buhmann
Yiannis Aloimonos	Stan Bileschi	Hans Burkhardt
Tal Arbel	Stan Birchfield	Andrew Calway
Kalle Åström	Volker Blanz	Rodrigo Carceroni
Peter Auer	Aaron Bobick	Gustavo Carneiro
Jonas August	Endre Boros	M. Carreira-Perpinan
Shai Avidan	Terrance Boult	Tat-Jen Cham
Simon Baker	Richard Bowden	Rama Chellappa
Kobus Barnard	Edmond Boyer	German Cheung
Adrien Bartoli	Yuri Boykov	Ondřej Chum
Benedicte Bascle	Gary Bradski	James Clark
Csaba Beleznai	Chris Bregler	Isaac Cohen
Peter Belhumeur	Thomas Breuel	Laurent Cohen
Serge Belongie	Gabriel Brostow	Michael Cohen
Moshe Ben-Ezra	Matthew Brown	Robert Collins
Alexander Berg	Michael Brown	Dorin Comaniciu

James Coughlan
David Crandall
Daniel Cremers
Antonio Criminisi
David Cristinacce
Gabriela Csurka
Navneet Dalal
Kristin Dana
Kostas Daniilidis
Larry Davis
Andrew Davison
Nando de Freitas
Daniel DeMenthon
David Demirdjian
Joachim Denzler
Michel Dhome
Sven Dickinson
Gianfranco Doretto
Gyuri Dorko
Pinar Duygulu Sahin
Charles Dyer
James Elder
Irfan Essa
Andras Ferencz
Rob Fergus
Vittorio Ferrari
Sanja Fidler
Mario Figueiredo
Graham Finlayson
Robert Fisher
François Fleuret
Wolfgang Förstner
Charless Fowlkes
Jan-Michael Frahm
Friedrich Fraundorfer
Bill Freeman
Brendan Frey
Andrea Frome
Pascal Fua
Yasutaka Furukawa
Daniel Gatica-Perez
Dariu Gavrila
James Gee
Guido Gerig
Theo Gevers
Christopher Geyer
Michael Goesele
Dan Goldman
Shaogang Gong
Leo Grady
Kristen Grauman
Eric Grimson
Fred Hamprecht
Edwin Hancock
Allen Hanson
James Hays
Carlos Hernández
Anders Heyden
Adrian Hilton
David Hogg
Derek Hoiem
Alex Holub
Anthony Hoogs
Daniel Huber
Alexander Ihler
Michal Irani
Hiroshi Ishikawa
David Jacobs
Bernd Jähne
Hervé Jégou
Ian Jermyn
Nebojsa Jojic
Michael Jones
Frédéric Jurie
Timor Kadir
Fredrik Kahl
Amit Kale
Kenichi Kanatani
Sing Bing Kang
Robert Kaucic
Qifa Ke
Renaud Keriven
Charles Kervrann
Ron Kikinis
Benjamin Kimia
Ron Kimmel
Josef Kittler
Hedvig Kjellström
Leif Kobbelt
Pushmeet Kohli
Esther Koller-Meier
Vladimir Kolmogorov
Nikos Komodakis
Kurt Konolige
Jana Košecká
Zuzana Kukelova
Sanjiv Kumar
Kyros Kutulakos
Ivan Laptev
Longin Jan Latecki
Svetlana Lazebnik
Erik Learned-Miller
Yann Lecun
Bastian Leibe
Vincent Lepetit
Thomas Leung
Anat Levin
Fei-Fei Li
Hongdong Li
Stephen Lin
Jim Little
Ce Liu
Yanxi Liu
Brian Lovell
Simon Lucey
John Maccormick
Petros Maragos
Aleix Martinez
Iain Matthews
Wojciech Matusik
Bruce Maxwell
Stephen Maybank
Stephen McKenna
Peter Meer
Etienne Mémin
Dimitris Metaxas
Branislav Mičušík
Krystian Mikolajczyk
Anurag Mittal
Theo Moons
Greg Mori
Pawan Mudigonda
David Murray
Srinivasa Narasimhan
Randal Nelson

Ram Nevatia
Jean-Marc Odobez
Björn Ommer
Nikos Paragios
Vladimir Pavlovic
Shmuel Peleg
Marcello Pelillo
Pietro Perona
Maria Petrou
Vladimir Petrovic
Jonathon Phillips
Matti Pietikäinen
Axel Pinz
Robert Pless
Tom Pock
Fatih Porikli
Simon Prince
Long Quan
Ravi Ramamoorthi
Deva Ramanan
Anand Rangarajan
Ramesh Raskar
Xiaofeng Ren
Jens Rittscher
Rómer Rosales
Bodo Rosenhahn
Peter Roth
Stefan Roth
Volker Roth
Carsten Rother
Fred Rothganger
Daniel Rueckert
Dimitris Samaras
Radim Šára
Eric Saund
Silvio Savarese
Daniel Scharstein
Yoav Schechner
Kondrad Schindler
Stan Sclaroff
Mubarak Shah
Gregory Shakhnarovich
Eli Shechtman
Jianbo Shi
Kaleem Siddiqi
Leonid Sigal
Sudipta Sinha
Josef Sivic
Cristian Sminchişescu
Anuj Srivastava
Drew Steedly
Gideon Stein
Björn Stenger
Christoph Strecha
Erik Sudderth
Josephine Sullivan
David Suter
Tomáš Svoboda
Hai Tao
Marshall Tappen
Demetri Terzopoulos
Carlo Tomasi
Fernando Torre
Lorenzo Torresani
Emanuele Trucco
David Tschumperlé
John Tsotsos
Peter Tu
Matthew Turk
Oncel Tuzel
Carole Twining
Ranjith Unnikrishnan
Raquel Urtasun
Joost Van de Weijer
Manik Varma
Nuno Vasconcelos
Olga Veksler
Jakob Verbeek
Luminita Vese
Thomas Vetter
René Vidal
George Vogiatzis
Daphna Weinshall
Michael Werman
Tomáš Werner
Richard Wildes
Lior Wolf
Ying Wu
Eric Xing
Yaser Yacoob
Ruigang Yang
Stella Yu
Lihi Zelnik-Manor
Richard Zemel
Li Zhang
S. Zhou
Song-Chun Zhu
Todd Zickler
Lawrence Zitnick

Additional Reviewers

Lourdes Agapito
Daniel Alexander
Elli Angelopoulou
Alexandru Balan
Adrian Barbu
Nick Barnes
João Barreto
Marian Bartlett
Herbert Bay
Ross Beveridge
V. Bhagavatula
Edwin Bonilla
Aeron Buchanan
Michael Burl
Tiberio Caetano
Octavia Camps
Sharat Chandran
François Chaumette
Yixin Chen
Dmitry Chetverikov
Sharat Chikkerur
Albert Chung
Nicholas Costen
Gabriela Oana Cula
Goksel Dedeoglu
Hervé Delingette
Michael Donoser

Mark Drew
Zoran Duric
Wolfgang Einhauser
Aly Farag
Beat Fasel
Raanan Fattal
Paolo Favaro
Rogerio Feris
Cornelia Fermüller
James Ferryman
David Forsyth
Jean-Sébastien Franco
Mario Fritz
Andrea Fusiello
Meirav Galun
Bogdan Georgescu
A. Georghiades
Georgy Gimel'farb
Roland Goecke
Toon Goedeme
Jacob Goldberger
Luis Goncalves
Venu Govindaraju
Helmut Grabner
Michael Grabner
Hayit Greenspan
Etienne Grossmann
Richard Harvey
Sam Hasinoff
Horst Haussecker
Jesse Hoey
Slobodan Ilic
Omar Javed
Qiang Ji
Jiaya Jia
Hailin Jin
Ioannis Kakadiaris
Joni-K. Kämäräinen
George Kamberov
Yan Ke
Andreas Klaus
Georg Klein
Reinhard Koch
Mathias Kolsch
Andreas Koschan
Christoph Lampert
Mike Langer
Georg Langs
Neil Lawrence
Sang Lee
Boudewijn Lelieveldt
Marc Levoy
Michael Lindenbaum
Chengjun Liu
Qingshan Liu
Manolis Lourakis
Ameesh Makadia
Ezio Malis
R. Manmatha
David Martin
Daniel Martinec
Yasuyuki Matsushita
Helmut Mayer
Christopher Mei
Paulo Mendonça
Majid Mirmehdi
Philippos Mordohai
Pierre Moreels
P.J. Narayanan
Nassir Navab
Jan Neumann
Juan Carlos Niebles
Ko Nishino
Thomas O'Donnell
Takayuki Okatani
Kenji Okuma
Margarita Osadchy
Mustafa Ozuysal
Sharath Pankanti
Sylvain Paris
James Philbin
Jean-Philippe Pons
Emmanuel Prados
Zhen Qian
Ariadna Quattoni
Ali Rahimi
Ashish Raj
Visvanathan Ramesh
Christopher Rasmussen
Tammy Riklin-Raviv
Charles Rosenberg
Arun Ross
Michael Ross
Szymon Rusinkiewicz
Bryan Russell
Sudeep Sarkar
Yoichi Sato
Ashutosh Saxena
Florian Schroff
Stephen Se
Nicu Sebe
Hans-Peter Seidel
Steve Seitz
Thomas Serre
Alexander Shekhovtsov
Ilan Shimshoni
Michal Sofka
Jan Solem
Gerald Sommer
Jian Sun
Rahul Swaminathan
Hugues Talbot
Chi-Keung Tang
Xiaoou Tang
C.J. Taylor
Jean-Philippe Thiran
David Tolliver
Yanghai Tsin
Zhuowen Tu
Vaibhav Vaish
Anton van den Hengel
Bram Van Ginneken
Dirk Vandermeulen
Alessandro Verri
Hongcheng Wang
Jue Wang
Yizhou Wang
Gregory Welch
Ming-Hsuan Yang
Caspi Yaron
Jieping Ye
Alper Yilmaz
Christopher Zach
Hongyuan Zha
Cha Zhang
Jerry Zhu
Lilla Zollei

Sponsoring Institutions

Table of Contents – Part I

Lecture by Prof. Jan Koenderink

Recognition

Poster Session I

Stereo

Table of Contents – Part II

People

Faces

Poster Session II

Tracking

Table of Contents – Part III

Matching

Learning+Features

Poster Session III

MRFs

Table of Contents – Part IV

Segmentation

Computational Photography

Poster Session IV

Active Reconstruction

Something Old, Something New, Something Borrowed, Something Blue

Jan J. Koenderink

EEMCS, Delft University of Technology, The Netherlands

Abstract. My first paper of a "Computer Vision" signature (on invariants related to optic flow) dates from 1975. I have published in Computer Vision (next to work in cybernetics, psychology, physics, mathematics and philosophy) till my retirement earlier this year (hence the slightly blue feeling), thus my career roughly covers the history of the field. "Vision" has diverse connotations. The fundamental dichotomy is between "optically guided action" and "visual experience". The former applies to much of biology and computer vision and involves only concepts from science and engineering (e.g., "inverse optics"), the latter involves intention and meaning and thus additionally involves concepts from psychology and philosophy. David Marr's notion of "vision" is an uneasy blend of the two: On the one hand the goal is to create a "representation of the scene in front of the eye" (involving intention and meaning), on the other hand the means by which this is attempted are essentially "inverse optics". Although this has nominally become something of the "Standard Model" of CV, it is actually incoherent. It is the latter notion of "vision" that has always interested me most, mainly because one is still grappling with basic concepts. It has been my aspiration to turn it into science, although in this I failed. Yet much has happened (something old) and is happening now (something new). I will discuss some of the issues that seem crucial to me, mostly illustrated through my own work, though I shamelessly borrow from friends in the CV community where I see fit.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, p. 1, 2008.

Learning to Localize Objects with Structured Output Regression

Matthew B. Blaschko and Christoph H. Lampert

Max Planck Institute for Biological Cybernetics
72076 Tübingen, Germany
{blaschko,chl}@tuebingen.mpg.de

Abstract. Sliding window classifiers are among the most successful and widely applied techniques for object localization. However, training is typically done in a way that is not specific to the localization task. First a binary classifier is trained using a sample of positive and negative examples, and this classifier is subsequently applied to multiple regions within test images. We propose instead to treat object localization in a principled way by posing it as a problem of *predicting structured data*: we model the problem not as binary classification, but as the prediction of the bounding box of objects located in images. The use of a *joint-kernel* framework allows us to formulate the training procedure as a generalization of an SVM, which can be solved efficiently. We further improve computational efficiency by using a branch-and-bound strategy for localization during both training and testing. Experimental evaluation on the PASCAL VOC and TU Darmstadt datasets show that the structured training procedure improves performance over binary training as well as the best previously published scores.

1 Introduction

Object localization, also called *object detection*, is an important task for image understanding, *e.g.* in order to separate an object from the background, or to analyze the spatial relations of different objects. Object localization is commonly performed using sliding window classifiers [1,2,3,4,5,6]. Sliding window classifiers train a discriminant function and then scan over locations in the image, often at multiple scales, and predict that the object is present in subwindows with high score. This approach has been shown to be very effective in many situations, but suffers from two main disadvantages: (i) it is computationally inefficient to scan over the entire image and test every possible object location, and (ii) it is not clear how to optimally train a discriminant function for localization. The first issue has been recently addressed in [7] by using a branch-and-bound optimization strategy to efficiently determine the bounding box with the maximum score of the discriminant function. We address the second issue in this work by proposing a training strategy that specifically optimizes localization accuracy, resulting in much higher performance than systems that are trained, *e.g.*, using a support vector machine.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 2–15, 2008.

In particular, we utilize a machine learning approach called *structured learning*. Structured learning is the problem of learning to predict outputs that are not simple binary labels, but instead have a more complex structure. By appropriately modeling the relationships between the different outputs within the output space, we can learn a classifier that efficiently makes better use of the available training data. In the context of object localization, the output space is the space of possible bounding boxes, which can be parameterized, *e.g.*, by four numbers indicating the top, left, right, and bottom coordinates of the region. The coordinates can take values anywhere between 0 and the image size, thus making the setup a problem of *structured regression* rather than classification. Furthermore, the outputs are not independent of each other; the right and bottom coordinates have to be larger than the top and bottom coordinates, and predicting the top of the box independently of the left of the box will almost certainly give worse results than predicting them together. Additionally, the score of one possible bounding box is related to the scores of other bounding boxes; two highly overlapping bounding boxes will have similar objectives. By modeling the problem appropriately, we can use these dependencies to improve performance and efficiency of both the training and testing procedures.

The rest of the paper is organized as follows. In Section 2 we discuss previous work in object localization and structured learning and its relation to the proposed method. In Section 3 we introduce the optimization used to train our structured prediction model. The loss function is presented in Section 3.1, while a joint kernel map for object localization is presented in Section 3.2. We discuss a key component of the optimization in Section 4. Experimental results are presented in Section 5 and discussed in Section 6. Finally, we conclude in Section 7.

2 Related Work

Localization of arbitrary object classes has been approached in many ways in the literature. Constellation models detect object parts and the relationship between them. They have been trained with varying levels of supervision and with both generative and discriminative approaches [8,9,10]. A related approach has been to use object parts to vote for the object center and then search for maxima in the voting process using a generalized Hough transform [11]. This approach has also been combined with a discriminatively trained classifier to improve performance [12]. Alternatively, [13] have taken the approach of computing image segments in an unsupervised fashion and cast the localization problem as determining whether each of the segments is an instance of the object. Sliding window classifiers are among the most popular approaches to object localization [1,2,3,4,5,6,7], and the work presented in this paper can broadly be seen to fall into this category. The sliding window approach consists of training a classifier, *e.g.* using neural networks [5], boosted cascades of features [6], exemplar models [2,7], or support vector machines [1,3,4,7], and then evaluating the trained classifier at various locations in the image. Each of these techniques rely

on finding modes of the classifier function in the image, and then generally use a non-maximal suppression step to avoid multiple detections of the same object. This of course requires on a classifier function that has modes at the location of objects and not elsewhere. However, while discriminatively trained classifiers generally have high objectives at the object location, they are not specifically trained for this property and the modes may not be well localized. One approach to address this problem is to train a classifier iteratively in a boosting fashion: after each step, localization mistakes are identified and added to the training data for the next iteration, *e.g.* [3,5]. These techniques, however, cannot handle the case when earlier iterations partially overlap with the true object because incorporating these locations would require either an overlap threshold or fractional labels. In contrast, we propose an approach that uses *all* bounding boxes as training examples and that handles partial detections by appropriately scaling the classifier loss. As we show in subsequent sections, we can efficiently take advantage of the structure of the problem to significantly improve results by using this localization specific training.

3 Object Localization as Structured Learning

Given a set of input images $\{x_1, \ldots, x_n\} \subset \mathcal{X}$ and their associated annotations $\{y_1, \ldots, y_n\} \subset \mathcal{Y}$, we wish to learn a mapping $g : \mathcal{X} \mapsto \mathcal{Y}$ with which we can automatically annotate unseen images. We consider the case where the output space consists of a label indicating whether an object is present, and a vector indicating the top, left, bottom, and right of the bounding box within the image: $\mathcal{Y} \equiv \{(\omega, t, l, b, r) \mid \omega \in \{+1, -1\},\ (t, l, b, r) \in \mathbb{R}^4\}$. For $\omega = -1$ the coordinate vector (t, l, b, r) is ignored. We learn this mapping in the structured learning framework [14,15] as

$$g(x) = \operatorname{argmax}_y f(x, y) \tag{1}$$

where $f(x, y)$ is a discriminant function that should give a large value to pairs (x, y) that are well matched. The task is therefore to learn the function f, given that it is in a form that the maximization in Equation (1) can be done feasibly. We address the issue of maximization in Section 4.

To train the discriminant function, f, we use the following generalization of the support vector machine [14]

$$\min_{w,\xi} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \tag{2}$$

$$\text{s.t. } \xi_i \geq 0, \quad \forall i \tag{3}$$

$$\langle w, \phi(x_i, y_i)\rangle - \langle w, \phi(x_i, y)\rangle \geq \Delta(y_i, y) - \xi_i, \quad \forall i, \forall y \in \mathcal{Y} \setminus y_i \tag{4}$$

where $f(x_i, y) = \langle w, \phi(x_i, y)\rangle$, $\phi(x_i, y)$ is a joint kernel map implicitly defined by the kernel identity $k\left((x, y), (x', y')\right) = \langle \phi(x, y), \phi(x', y')\rangle$,

$$w = \sum_{i=1}^{n} \sum_{y \in \mathcal{Y} \setminus y_i} \alpha_{iy} \left(\phi(x_i, y_i) - \phi(x_i, y)\right), \tag{5}$$

and $\Delta(y_i, y)$ is a loss function that decreases as a possible output, y, approaches the true output, y_i. This optimization is convex and, given appropriate definitions of $\phi(x_i, y)$ and $\Delta(y_i, y)$, does not significantly differ from the usual SVM primal formulation except that there are an infeasibly large number of constraints in Equation (4) (the number of training samples times the size of the output space, which can even become infinite, *e.g.* in the case of continuous outputs). We note, however, that not all constraints will be active at any time, which can be seen by the equivalence between Equation (4) and

$$\xi_i \geq \max_{y \in \mathcal{Y} \setminus y_i} \Delta(y_i, y) - (\langle w, \phi(x_i, y_i) \rangle - \langle w, \phi(x_i, y) \rangle), \quad \forall i \tag{6}$$

which indicates that the α_{iy} in Equation (5) will be sparse. At training time, we can use *constraint generation* to solve the optimization in Equations (2)–(4). Estimates of w are trained using fixed subsets of constraints, and new constraints are added by finding the y that maximize the right hand side of Equation (6). This alternation is repeated until convergence, generally with a small set of constraints compared to the size of $\mathcal{Y}$. We therefore can efficiently optimize the discriminant function, f, given a choice of the loss $\Delta(y_i, y)$ and the kernel $k((x, y), (x', y'))$, as well as a method of performing the maximization in Equation (6). We discuss the loss function in Section 3.1, while we discuss the joint kernel in Section 3.2. A branch-and-bound procedure for the maximization step is explained in Section 4.

3.1 Choice of Loss Function

The choice of loss function $\Delta(y_i, y)$ should reflect the quantity that measures how well the system performs. We have chosen the following loss, which is constructed from the measure of *area overlap* used in the VOC challenges [16,17,18]

$$\Delta(y_i, y) = \begin{cases} 1 - \frac{\text{Area}(y_i \bigcap y)}{\text{Area}(y_i \bigcup y)} & \text{if } y_{i\omega} = y_\omega = 1 \\ 1 - \left(\frac{1}{2}(y_{i\omega} y_\omega + 1)\right) & \text{otherwise} \end{cases} \tag{7}$$

where $y_{i\omega} \in \{-1, +1\}$ indicates whether the object is present or absent in the image. $\Delta(y_i, y)$ has the desirable property that it is equal to zero in the case that the bounding boxes given by y_i and y are identical, and is 1 if they are disjoint. It also has several favorable properties compared to other possible object localization metrics [19], *e.g.* invariance to scale and translation. The formulation (7) is attractive in that it scales smoothly with the degree of overlap between the solutions, which is important to allow the learning process to utilize partial detections for training. In the case that y_i or y indicate that the object is not present in the image, we have a loss of 0 if the labels agree, and 1 if they disagree, which yields the usual notion of margin for an SVM. This setup automatically enforces by a maximum margin approach two conditions that are important for localization. First, in images that contain the object to be detected, the localized region should have the highest score of all possible boxes. Second, in images that do not contain the objects, no box should get a high score.

3.2 A Joint Kernel Map for Localization

To define the joint kernel map, $\phi(x_i, y)$, we note that kernels between images generally are capable of comparing images of differing size [1,4,20,21]. Cropping a region of an image and then applying an image kernel is a simple and elegant approach to comparing image regions. We use the notation $x|_y$ to denote the region of the image contained within the bounding box defined by y, and $\phi_x(x|_y)$ to denote the representation of $x|_y$ in the Hilbert space implied by a kernel over images, $k_x(\cdot,\cdot)$. If y indicates that the object is not present in the image, we consider $\phi_x(x|_y)$ to be equal to the $\mathbf{0}$ vector in the Hilbert space, *i.e.* for all x', $k_x(x|_y, x') = 0$. The resulting joint kernel map for object localization is therefore

$$k((x,y),(x',y')) = k_x(x|_y, x'|_{y'}). \tag{8}$$

Image kernels generally compute statistics or features of the two images and then compare them. This includes for example, bag of visual words methods [22], groups of contours [4], spatial pyramids [1,21], and histograms of oriented gradients [3]. An important property of the joint kernel defined in Equation (8) is that overlapping image regions will have common features and related statistics. This relationship can be exploited for computational efficiency, as we outline in the subsequent section.

4 Maximization Step

Since the maximization in Equation (6) has to be repeated many times during training, as well as a similar maximization at test time (Equation (1)), it is important that we can compute this efficiently. Specifically, at training time we need to compute

$$\begin{aligned} &\max_{y \in \mathcal{Y} \setminus y_i} \Delta(y_i, y) + \langle w, \phi(x_i, y) \rangle \\ &= \max_{y \in \mathcal{Y} \setminus y_i} \Delta(y_i, y) + \sum_{j=1}^{n} \sum_{\tilde{y} \in \mathcal{Y}} \alpha_{j\tilde{y}} \left(k_x(x_j|_{y_j}, x_i|_y) - k_x(x_j|_{\tilde{y}}, x_i|_y) \right) \end{aligned} \tag{9}$$

We therefore need an algorithm that efficiently maximizes

$$\max_{\substack{y \in \mathcal{Y} \setminus y_i \\ y_\omega = y_{i\omega} = 1}} -\frac{\text{Area}(y_i \bigcap y)}{\text{Area}(y_i \bigcup y)} + \sum_{j=1}^{n} \sum_{\tilde{y} \in \mathcal{Y}} \alpha_{j\tilde{y}} \left(k_x(x_j|_{y_j}, x_i|_y) - k_x(x_j|_{\tilde{y}}, x_i|_y) \right) \tag{10}$$

and for testing, we need to maximize the reduced problem

$$\max_{\substack{y \in \mathcal{Y} \\ y_\omega = 1}} \sum_{j=1}^{n} \sum_{\tilde{y} \in \mathcal{Y}} \alpha_{j\tilde{y}} \left(k_x(x_j|_{y_j}, x_i|_y) - k_x(x_j|_{\tilde{y}}, x_i|_y) \right) \tag{11}$$

The maximizations in Equations (10) and (11) can both be solved using a sliding window approach. In Equation (10), the maximization finds the location in the image that has simultaneously a high score for the given estimate of w and a

high loss (*i.e.* low overlap with ground truth). This is a likely candidate for a misdetection, and the system therefore considers it as a training constraint with the margin scaled to indicate how far the estimate is from ground truth. Because of the infeasible computational costs involved in an exhaustive search, sliding window approaches only evaluate the objective over a subset of possible bounding boxes and therefore give only an approximate solution to Equation (9). This can be viewed as searching for solutions in a strongly reduced set $\hat{\mathcal{Y}} \subset \mathcal{Y}$, where $\hat{\mathcal{Y}}$ includes only the bounding boxes that are evaluated in the sliding window search. However, we can it is more efficient to use a branch-and-bound optimization strategy as in [7], which gives the maximum over the entire set, $\mathcal{Y}$. We adapt this approach here to the optimization problems in Equations (10) and (11).

The branch and bound strategy consists of keeping a priority queue of sets of bounding boxes, which is ordered by an upper bound on the objective function. The algorithm is guaranteed to converge to the globally optimal solution provided the upper bound is equal to the true value of the quantity to be optimized when the cardinality of the set of bounding boxes is equal to one. The sets of bounding boxes, $\tilde{\mathcal{Y}}$, are represented compactly by minimum and maximum values of the top, left, bottom, and right coordinates of a bounding box. This procedure is fully specified given bounding functions, $\hat{h}$, for the objectives in Equations (10) and (11) (Algorithm 1).

Algorithm 1. Branch-and-Bound Optimization Procedure

Require: image $I \in \mathbb{R}^{n \times m}$
Require: quality bounding function $\hat{h}$
Ensure: $y = \text{argmax}_{R \subset I} f(R)$
  initialize P as empty priority queue
  initialize $\tilde{\mathcal{Y}} = [0, n] \times [0, m] \times [0, n] \times [0, m]$ indicating the top, left, bottom, and right of the box could fall anywhere in I
  repeat
    split $\tilde{\mathcal{Y}} \rightarrow \tilde{\mathcal{Y}}_1 \cup \tilde{\mathcal{Y}}_2$ by splitting the range of one of the sides into two
    push $(\hat{h}(\tilde{\mathcal{Y}}_1), \tilde{\mathcal{Y}}_1)$ and $(\hat{h}(\tilde{\mathcal{Y}}_2), \tilde{\mathcal{Y}}_2)$ into P
    retrieve top state, $\tilde{\mathcal{Y}}$, from P
  until $\tilde{\mathcal{Y}}$ consists of only one rectangle, y

We note that Equation (11) is simply a linear combination of kernel evaluations between $x_i|_y$ and the support vectors, and therefore is in exactly the form that was solved for in [7]. Bounds were given for a variety of kernels commonly used in the literature for image classification, while bounds for arbitrary kernels can be constructed using interval arithmetic [7]. Similarly, Equation (10) can be bounded by the sum of the bound for Equation (11) and a bound for the overlap term

$$\forall \tilde{y} \in \tilde{\mathcal{Y}}, -\frac{\text{Area}(y_i \bigcap \tilde{y})}{\text{Area}(y_i \bigcup \tilde{y})} \leq -\frac{\min_{y \in \tilde{\mathcal{Y}}} \text{Area}(y_i \bigcap y)}{\max_{y \in \tilde{\mathcal{Y}}} \text{Area}(y_i \bigcup y)}. \tag{12}$$

5 Evaluation

For evaluation we performed experiments on two publicly available computer vision datasets for object localization: TU Darmstadt `cows` and PASCAL VOC 2006 (Figures 1 and 2).

Fig. 1. Example images from the TU Darmstadt `cow` dataset. There is always exactly one cow in every image, but backgrounds vary.

Fig. 2. Example images from the PASCAL VOC 2006 dataset. Images can contain multiple object classes and multiple instances per class.

5.1 Experimental Setup

For both datasets we represent images by sets of local SURF descriptors [23] that are extracted from feature point locations on a regular grid, on salient points and on randomly chosen locations. We sample 100,000 descriptors from training images and cluster them using K-means into a 3,000-entry visual codebook. Subsequently, all feature points in train and test images are represented by their coordinates in the image and the ID of the corresponding codebook entry. Similar representations have been used successfully in many scenarios for object and scene classification [1,2,7,21,22].

To show the performance of the proposed *structured training* procedure, we benchmark it against *binary training*, which is a commonly used method to obtain a quality function for sliding window object localization [1,2,3,4,5,6,7]. It relies on first training a binary classifier and then using the resulting real-valued classifier function as quality function. As positive training data, one uses the ground truth object boxes. Since localization datasets usually do not contain boxes with explicitly negative class label, one samples boxes from background regions to use as the negative training set. In our setup, we implement this sampling in a way that ensures that negative boxes do not overlap with ground truth boxes or each other by more than 20%. The *binary training* consists of

training an SVM classifier with a kernel that is the linear scalar product of the *bag-of-visual-words* histograms. The SVM's regularization parameter C and number of negative boxes to sample per image are free parameters.

Our implementation of the proposed *structured training* makes use of the `SVMstruct` [14] package. It uses a *constraint generation* technique as explained in Section 3 to solve the optimization problem (2). This requires iterative identification of example-label pairs that most violate the constraints (6). We solve this by adapting the public implementation of the branch-and-bound optimization `ESS` [7] to include the loss term Δ.[1] As in the case of binary training, we use a linear image kernel (8) over the space of bag-of-visual-word histograms. The C parameter in Equation (2) is the only free parameter of the resulting training procedure.

5.2 Results: TU Darmstadt Cows

The TU Darmstadt `cow` dataset consists of 111 training and 557 test images of side views of cows in front of different backgrounds, see Figure 1 [24]. The dataset is useful to measure pure localization performance, because each training and test image contains exactly one cow. For other datasets, performance is often influenced by the decision whether an object is present at all or not, which is the problem of classification, not of localization. We train the binary and the structured learning procedure as described in the previous section. First we perform 5-fold cross validation on the training set, obtaining the SVM's regularization parameter C between 10^{-4} and 10^{4} for both training procedures, and the number of negative boxes to sampled between 1 and 10 for the binary training.

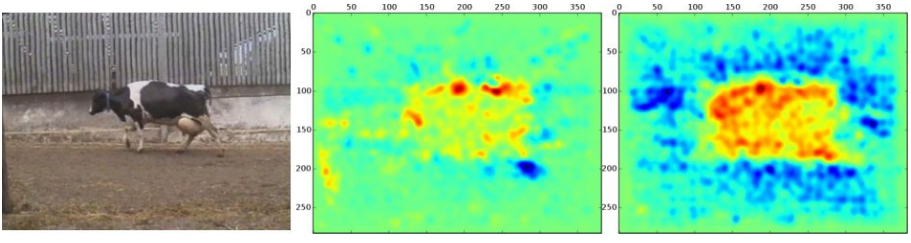


Fig. 3. Weight distribution for a TU Darmstadt `cow` test image (best viewed in color). Red indicates positive weights, blue indicates negative weights. Both methods assign positive weights to the cow area, but the structured learning better distributes them across the spatial extent. Additionally, structured learning better learns to give negative weight to image features that lie outside the object.

Afterwards, the systems are retrained on all images in the training set. The resulting systems are applied to the test set, which had not been used in any of the previous steps. We predict three possible object locations per image and

[1] The source code is available at the authors' homepages.

rank them by their detection score (Equation (1)). Figure 3 shows the resulting distribution of weights for an example image in the test set.

The object localization step detect in each image the rectangular region that maximizes the sum of scores, which is a 4-dimensional search space. We visualize the quality function with contour lines of different two-dimensional intersections through the parameter space (Figure 4). The left block of plots shows the quality function for the upper left corner when we fix the lower right corner of the detection box to the ground truth location. The right block shows the quality for the box center when fixing the box dimensions to their ground truth size. Structured training achieves tighter contours, indicating a stronger maximum of the quality function at the correct location.

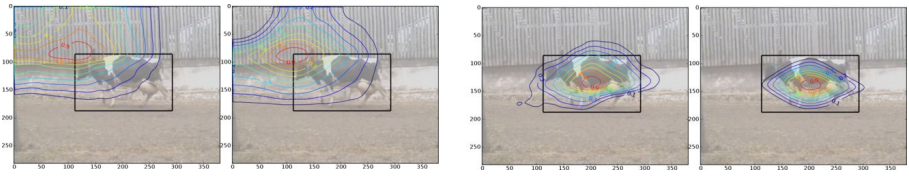


Fig. 4. Contour plots of the learned quality function for a TU Darmstadt `cow` test image (best viewed in color). The first and third image corresponds to the quality function learned by binary training, the second and fourth image shows structured training. In left block shows the quality of the upper left corner when fixing the bottom right corner at its ground truth coordinates. The right block shows the quality of the center point when keeping the box dimensions fixed at their ground truth values. Structured learning achieves tighter contours, indicating less uncertainty in localization.

This effect is also shown numerically: we calculate precision–recall curves using the overlap between detected boxes and ground truth as the criterion for correct detections (for details see [12]). Table 1 contains the performance at the point of equal-error rate. The structured detection framework achieves performance superior to binary training and to the previously published methods.

Table 1. Performance on TU Darmstadt `cows` dataset at equal error rate. *Binary training* achieves result on par with the best previously reported *implicit shape model (ISM)*, *local kernels (LK)* and their combination *(LK+ISM)* [12]. *Structured training* improves over the previous methods.

	ISM	LK	LK+ISM	binary training	structured training
performance at EER	96.1%	95.3%	97.1%	97.3%	**98.2%**

5.3 Results: PASCAL VOC 2006

The PASCAL VOC 2006 dataset [17] contains 5,304 images of 10 object classes, evenly split into a *train/validation* and a *test* part. The images were mostly downloaded from the internet and then used for the PASCAL challenge on Visual

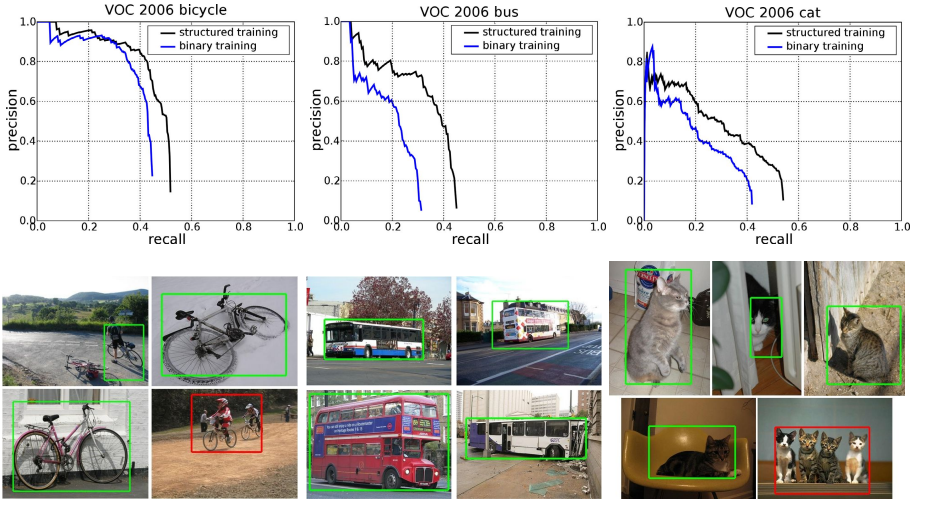


Fig. 5. Precision–recall curves and example detections for the PASCAL VOC `bicycle`, `bus` and `cat` category (from left to right). Structured training improves both, precision and recall. Red boxes are counted as mistakes by the VOC evaluation routine, because they are too large or contain more than one object.

Object Categorization in 2006. The dataset contains ground truth in the form of bounding boxes that were generated manually. Since the images contain natural scenes, many contain more than one object class or several instances of the same class. Evaluation is performed based on precision-recall curves for which the system returns a set of candidate boxes and confidence scores for every object category. Detected boxes are counted as correct if their area overlap with a ground truth box exceeds 50% [17].

We use the binary and the structured procedures to train localization systems for all 10 categories. Parameter selection is done separately for each class, choosing the parameter C and number of boxes to sampled based on the performance when trained on the *train* and evaluated on the *val* part of the data. The range of parameters is identical to the TU Darmstadt `cow` dataset. The resulting system is then retrained on the whole *train/val* portion, excluding those which are marked as *difficult* in the ground truth annotation. For the *structured training*, we only train on the training images that contained the object to be detected, while for the *binary training* negative image regions were sampled from images with and without the object present.

The VOC dataset is strongly unbalanced, and in per-class object detection, most test images do not contain the objects to be detected at all. This causes the sliding window detection scores to become an unreliable measure for ranking. Instead, we calculate confidence scores for each detection from the output of a separate SVM with χ^2-kernel, based on the image and box cluster histograms. The relative weight between box and image kernel is determined by cross-validation. The same resulting classifier is used to rank the detection outputs of both training methods.

Table 2. Average Precision (AP) scores on the 10 categories of PASCAL VOC 2006. Structured training consistently improves over binary training, achieving 5 new best scores. In one category binary training achieves better results than structured training, but both methods improve the state-of-the-art. Results best in competition were reported in [17]. Results post competition were published after the official competition: †[25], ‡[2], *[7], $^{+}$[10].

	bike	bus	car	cat	cow	dog	horse	m.bike	person	sheep
structured training	.472	**.342**	.336	**.300**	**.275**	.150	**.211**	**.397**	.107	.204
binary training	.403	.224	.256	.228	.114	**.173**	.137	.308	.104	.099
best in competition	.440	.169	.444	.160	.252	.118	.140	.390	.164	**.251**
post competition	**.498**†	.249‡	**.458**†	.223*	—	.148*	—	—	**.340**$^{+}$	—

Figure 5.3 shows the resulting precision–recall curves on the test data for 3 of the categories. For illustration, we also show some example detections of the detection system based on structured learning. From the curves we can see that structured training improves both precision and recall of the detection compared to the binary training. Table 2 summarizes the results in numerical form using the *average precision* (AP) evaluation that was also used in the original VOC challenge. For reference, we also give the results of the best results in the 2006 challenge and the best results from later publications. Object localization with structured training achieves new best scores for 5 of the 10 categories. In all but one category, it achieved better results than the binary training, often by a large margin. In the remaining category, binary training obtains a better score, but in fact both training methods improve over the previous state-of-the-art.

6 Discussion

We have seen in the previous sections that the structured training approach can improve the quality of object detection in a sliding window setup. Despite the simple choice of a single feature set and a linear image kernel, we achieve results that often exceed the state-of-the art. In the following we discuss several explanations for its high performance.

First, structured learning can make more efficient use of the possible training data, because it has access to *all* possible boxes in the input images. During the training procedure, it automatically identifies the relevant boxes and incorporates them into the training set, focusing the training on locations where mistakes would otherwise be made. This is in contrast to binary training in which the ground truth object boxes are used as positive examples and negative examples are sampled from background regions. The number of negative boxes is by necessity limited in order balance the training set and avoid degenerate classifiers. However, sampling negative regions prior to training is done "blindly," without knowing if the sampled boxes are at all informative for training.

A second explanation is based on the observation that machine learning techniques work best if the statistical sample distribution is the same during the

training phase as it is during the test phase. For the standard sliding window approach that has been trained as a binary classifier, this is not the case. The training set only contains examples that either completely show the object to be detected, or not at all. At test time, however, many image regions have to be evaluated that contain portions of the object. Since the system was not trained for such samples, one can only hope that the classifier function will not assign any modes to these regions. In contrast, structured training is able to appropriately handle partial detections by scaling the loss flexibly, depending on the degree of overlap to the true solution. Note that a similar effect cannot easily be achieved for a binary iterative procedure: even when iterating over the training set multiple times and identifying wrong detections, only completely false positive detections can be reinserted as negative examples to the training set and made use of in future iterations. Partial detections would require a training label that is neither $+1$ or -1, and binary classifiers are not easily adapted to this case.

7 Conclusions

We have proposed a new method for object localization in natural images. Our approach relies on a structured-output learning framework that combines the advantages of the well understood sliding window procedure with a novel training step that avoids prediction mistakes by implicitly taking into account all possible object locations in the input image.

The approach gives superior results compared with binary training because it uses a training procedure that specifically optimizes for the task of localization, rather than for classification accuracy on a training set. It achieves this in several ways. First, it is statistically efficient; by implicitly using all possible bounding boxes as training data, we can make better use of the available training images. Second, it appropriately handles partial detections in order to tune the objective function and ensure that the modes correspond exactly to object regions and is not distracted by features that may be discriminative but are not representative for the object as a whole.

The structured training procedure can be solved efficiently by constraint generation, and we further improve the computational efficiency of both training and testing by employing a branch-and-bound strategy to detect regions within the image that maximize the training and testing subproblems. The resulting system achieves excellent performance, as demonstrated by new best results on the TU Darmstadt `cow` and PASCAL VOC 2006 datasets.

In future work, we will adapt our implementation to different image kernels, and explore strategies for speeding up the training procedure. We have only explored a margin rescaling technique for incorporating the variable loss, while a promising alternate formulation would rely on slack rescaling. We plan an empirical evaluation of these alternatives, along with a comparison to related adaptive training techniques, *e.g.* bootstrapping or boosted cascades.

Acknowledgments

This work was funded in part by the EC project CLASS, IST 027978. The first author is supported by a Marie Curie fellowship under the EC project PerAct, EST 504321. We would like to thank Mario Fritz for making the *TU Darmstadt cow* dataset available to us.

References

1. Bosch, A., Zisserman, A., Muñoz, X.: Representing Shape with a Spatial Pyramid Kernel. In: CIVR (2007)
2. Chum, O., Zisserman, A.: An Exemplar Model for Learning Object Classes. In: CVPR (2007)
3. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: CVPR, pp. 886–893 (2005)
4. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of Adjacent Contour Segments for Object Detection. PAMI 30, 36–51 (2008)
5. Rowley, H.A., Baluja, S., Kanade, T.: Human Face Detection in Visual Scenes. In: NIPS, vol. 8, pp. 875–881 (1996)
6. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. CVPR 1, 511 (2001)
7. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Beyond Sliding Windows: Object Localization by Efficient Subwindow Search. In: CVPR (2008)
8. Fergus, R., Zisserman, P.P.A.: Weakly Supervised Scale-Invariant Learning of Models for Visual Recognition. IJCV 71, 273–303 (2007)
9. Bouchard, G., Triggs, B.: Hierarchical part-based visual object categorization. In: CVPR, Washington, DC, USA, pp. 710–715. IEEE Computer Society Press, Los Alamitos (2005)
10. Felzenszwalb, P., McAllester, D., Ramanan, D.: A Discriminatively Trained, Multiscale, Deformable Part Model. In: CVPR (2008)
11. Leibe, B., Leonardis, A., Schiele, B.: Combined Object Categorization and Segmentation with an Implicit Shape Model. In: Workshop on Statistical Learning in Computer Vision, Prague, Czech Republic (2004)
12. Fritz, M., Leibe, B., Caputo, B., Schiele, B.: Integrating representative and discriminative models for object category detection. In: ICCV, pp. 1363–1370 (2005)
13. Viitaniemi, V., Laaksonen, J.: Techniques for Still Image Scene Classification and Object Detection. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4132, pp. 35–44. Springer, Heidelberg (2006)
14. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: ICML, p. 104 (2004)
15. Bakır, G.H., Hofmann, T., Schölkopf, B., Smola, A.J., Taskar, B., Vishwanathan, S.V.N.: Predicting Structured Data. MIT Press, Cambridge (2007)
16. Everingham, M., et al.: The 2005 PASCAL Visual Object Classes Challenge. In: Selected Proceedings of the First PASCAL Challenges Workshop, pp. 117–176. Springer, Heidelberg (2006)
17. Everingham, M., Zisserman, A., Williams, C.K.I., Van Gool, L.: The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results (2006), `http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf`

18. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results (2007), http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
19. Hemery, B., Laurent, H., Rosenberger, C.: Comparative study of metrics for evaluation of object localisation by bounding boxes. In: ICIG, pp. 459–464 (2007)
20. Eichhorn, J., Chapelle, O.: Object Categorization with SVM: Kernels for Local Features. Technical Report 137, Max Planck Institute for Biological Cybernetics, Tübingen, Germany (2004)
21. Lazebnik, S., Schmid, C., Ponce, J.: Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In: CVPR, pp. 2169–2178 (2006)
22. Nowak, E., Jurie, F., Triggs, B.: Sampling strategies for bag-of-features image classification. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 490–503. Springer, Heidelberg (2006)
23. Bay, H., Tuytelaars, T., Van Gool, L.J.: SURF: Speeded Up Robust Features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
24. Magee, D.R., Boyle, R.D.: Detecting Lameness Using 'Re-Sampling Condensation' and 'Multi-Stream Cyclic Hidden Markov Models'. Image and Vision Computing 20, 581–594 (2002)
25. Crandall, D.J., Huttenlocher, D.P.: Composite models of objects and scenes for category recognition. In: CVPR (2007)

Beyond Nouns: Exploiting Prepositions and Comparative Adjectives for Learning Visual Classifiers*

Abhinav Gupta and Larry S. Davis

Department of Computer Science
University of Maryland, College Park
{agupta,lsd}@cs.umd.edu

Abstract. Learning visual classifiers for object recognition from weakly labeled data requires determining correspondence between image regions and semantic object classes. Most approaches use co-occurrence of "nouns" and image features over large datasets to determine the correspondence, but many correspondence ambiguities remain. We further constrain the correspondence problem by exploiting additional language constructs to improve the learning process from weakly labeled data. We consider both "prepositions" and "comparative adjectives" which are used to express relationships between objects. If the models of such relationships can be determined, they help resolve correspondence ambiguities. However, learning models of these relationships requires solving the correspondence problem. We simultaneously learn the visual features defining "nouns" and the differential visual features defining such "binary-relationships" using an EM-based approach.

1 Introduction

There has been recent interest in learning visual classifiers of objects from images with text captions. This involves establishing correspondence between image regions and semantic object classes named by the nouns in the text. There exist significant ambiguities in correspondence of visual features and object classes. For example, figure 1 contains an image which has been annotated with the nouns "car" and "street". It is difficult to determine which regions of the image correspond to which word unless additional images are available containing "street" but not "car" (and vice-versa). A wide range of automatic image annotation approaches use such co-occurrence relationships to address the correspondence problem.

Some words, however, almost always occur in fixed groups, which limits the utility of co-occurrence relationships, alone, to reduce ambiguities in correspondence. For example, since cars are typically found on streets, it is difficult to resolve the correspondence using co-occurrence relationships alone. While such

* The authors would like to thank Kobus Barnard for providing the Corel-5k dataset. The authors would also like to acknowledge VACE for supporting the research.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 16–29, 2008.

confusion is not a serious impediment for image annotation, it is a problem if localization is a goal[1].

We describe how to reduce ambiguities in correspondence by exploiting natural relationships that exists between objects in an image. These relationships correspond to language constructs such as "prepositions" (e.g. above, below) and "comparative adjectives" (e.g. brighter, smaller). If models for such relationships were known and images were annotated with them, then they would constrain the correspondence problem and help resolve ambiguities. For example, in figure 1, consider the binary relationship $on(car, street)$. Using this relationship, we can trivially infer that the green region corresponds to "car" and the magenta region corresponds to "street".

The size of the vocabulary of binary relationships is very small compared to the vocabulary of nouns/objects. Therefore, human knowledge could be tapped to specify rules which can act as classifiers for such relationships (for example, a binary relationship $above(s_1, p_1) \Rightarrow s_1.y < p_1.y$). Alternatively, models can be learned from annotated images. Learning such binary relationships from a weakly-labeled dataset would be "straight forward" if we had a solution to the correspondence problem at hand. This leads to a chicken-egg problem, where models for the binary relationships are needed for solving the correspondence problem, and the solution of the correspondence problem is required for acquiring models of the binary relationships. We utilize an EM-based approach to simultaneously learn visual classifiers of objects and "differential" models of common prepositions and comparative binary relationships.

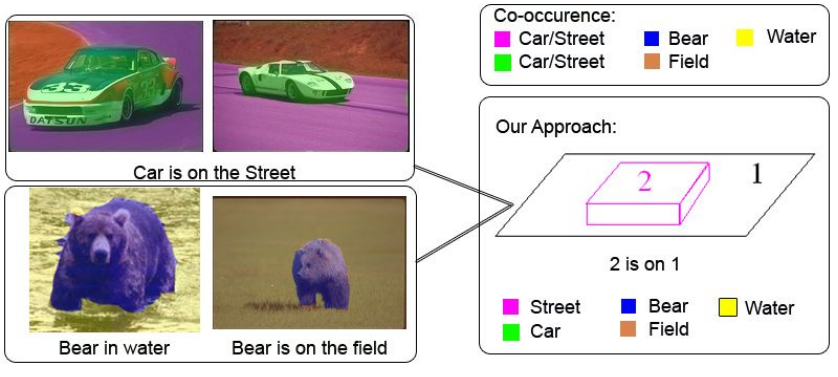


Fig. 1. An example of how our approach can resolve ambiguities. In the case of co-occurrence based approaches, it is hard to correspond the magenta/green regions to 'car'/'street'. 'Bear', 'water' and 'field' are easy to correspond. However, the correct correspondences of 'bear' and 'field' can be used to acquire a model for the relation 'on'. We can then use that model to classify the green region as belonging to 'car' and the magenta one to 'street', since only this assignment satisfies the binary relationship.

[1] It has also been argued [1] that for accurate retrieval, understanding image semantics (spatial localization) is critical.

The significance of the work is threefold: (1) It allows us to learn classifiers (i.e models) for a vocabulary of prepositions and comparative adjectives. These classifiers are based on differential features extracted from pairs of regions in an image. (2) Simultaneous learning of nouns and relationships reduces correspondence ambiguity and leads to better learning performance. (3) Learning priors on relationships that exist between nouns constrains the annotation problem and leads to better labeling and localization performance on the test dataset.

2 Related Work

Our work is clearly related to prior work on relating text captions and image features for automatic image annotation [2,3,4]. Many learning approaches have been used for annotating images which include translation models [5], statistical models [2,6], classification approaches [7,8,9] and relevance language models [10,11].

Classification based approaches build classifiers without solving the correspondence problem. These classifiers are learned on positive and negative examples generated from captions. Relevance language models annotate a test image by finding similar images in the training dataset and using the annotation words shared by them.

Statistical approaches model the joint distribution of nouns and image features. These approaches use co-occurrence counts between nouns and image features to predict the annotation of a test image [12,13]. Barnard et al. [6] presented a generative model for image annotation that induces hierarchical structure from the co-occurrence data. Srikanth et al. [14] proposed an approach to use the hierarchy induced by WordNet for image annotation. Duygulu et al. [5] modeled the problem as a standard machine translation problem. The image is assumed to be a collection of blobs (vocabulary of image features) and the problem becomes analogous to learning a lexicon from aligned bi-text. Other approaches such as [15] also model word to word correlations where prediction of one word induces a prior on prediction of other words.

All these approaches use co-occurrence relationships between nouns and image features; but they cannot, generally, resolve all correspondence ambiguities. They do not utilize other constructs from natural language and speech tagging approaches [16,17]. As a trivial example, given the annotation "pink flower" and a model of the adjective "pink", one would expect a dramatic reduction in the set of regions that would be classified as a flower in such an image. Other language constructs, such as "prepositions" or "comparative adjectives", which express relationships between two or more objects in the image, can also resolve ambiguities.

Our goal is to learn models, in the form of classifiers, for such language constructs. Ferrari et al. [18] presented an approach to learn visual attributes from a training dataset of positive and negative images using a generative model. However, collecting a dataset for all such visual attributes is cumbersome. Ideally we would like to use the original training dataset with captions to learn the appearance of

nouns/adjectives and also understand the meanings of common prepositions and comparative adjectives. Barnard et al. [19] presented an approach for learning adjectives and nouns from the same dataset. They treat adjectives similarly to nouns and use a two step process to learn the models. In the first step, they consider only adjectives as annotated text and learn models for them using a latent model. In the second step, they use the same latent model to learn nouns where learned models of adjectives are used to provide prior probabilities for labeling nouns. While such an approach might be applicable to learning models for adjectives, it cannot be applied to learning models for higher order(binary) relationships unless the models for the nouns are given.

Barnard et al. [20] also presented an approach to reduce correspondence ambiguity in weakly labeled data. They separate the problems of learning models of nouns from resolving correspondence ambiguities. They use a loose model for defining affinities between different regions and use the principal of exclusion reasoning to resolve ambiguities. On the other hand, we propose an approach to simultaneously resolve correspondence ambiguities and learn models of nouns using other language constructs which represent higher order relationships[2].

We also present a systematic approach to employing contextual information (second-order) for labeling images. The use of second order contextual information is very important during labeling because it can help resolve the ambiguities due to appearance confusion in many cases. For example, a blue homogeneous region, B, can be labeled as "water" as well as "sky" due to the similarity in appearance. However, the relation of the region to other nouns such as the "sun" can resolve the ambiguity. If the relation $below(B, sun)$ is more likely than $in(sun, B)$, then the region B can be labeled as "water" (and vice-versa). As compared to [20], which uses adjacency relations for resolution, our approach provides a broader range of relations(prepositions and comparative adjectives) that can be learned simultaneously with the nouns.

3 Overview

Each image in a training set is annotated with nouns and relationships between some subset of pairs of those nouns. We refer to each relationship instance, such as $above(A, B)$, as a predicate. Our goal is to learn classifiers for nouns and relationships (prepositions and comparative adjectives). Similar to [5], we represent each image with a set of image regions. Each image region is represented by a set of visual features based on appearance and shape (e.g area, RGB). The classifiers for nouns are based on these features. The classifiers for relationships are based on differential features extracted from pairs of regions such as the difference in area of two regions.

Learning models of both nouns and relationships requires assigning image regions to annotated nouns. As the data is weakly labeled, there is no explicit assignment of words to image regions. One could, however, assign regions to nouns

[2] The principles of exclusion reasoning are also applicable to our problem. We, however, ignore them here.

if the models of nouns and relationships were known. This leads to a chicken-egg problem. We treat assignment as the missing data and use an EM-approach to learn assignment and models simultaneously. In the E-step we evaluate possible assignments using the parameters obtained at previous iterations. Using the probabilistic distribution of assignment computed in the E-step, we estimate the maximum likelihood parameters of the classifiers in the M-step.

In the next section, we first discuss our model of generating predicates for a pair of image regions. This is followed by a discussion on learning the parameters of the model, which are the parameters of classifiers for nouns, prepositions and comparative adjectives.

4 Our Approach

4.1 Generative Model

We next describe the model for language and image generation for a pair of objects. Figure 2 shows our generative model.

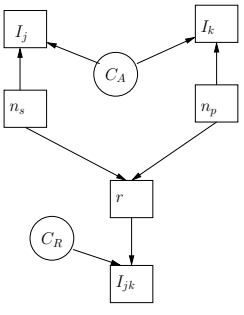


Fig. 2. The Graphical Model for Image Annotation

Each image is represented with a set of image regions and each region is associated with an object which can be classified as belonging to a certain semantic object class. These semantic object classes are represented by nouns in the vocabulary[3].

Assume two regions j and k are associated with objects belonging to semantic object classes, n_s and n_p respectively. Each region is described by a set of visual features I_j and I_k. The likelihood of image features I_j and I_k would depend on

[3] Generally, there will not be a one-one relationship between semantic object classes and nouns. For example, the word "bar" refers to two different semantic concepts in the sentences: "He went to the bar for a drink" and "There were bars in the window to prevent escape". Similarly, one semantic object class can be described by two or more words(synonyms). While dealing with synonyms and word sense disambiguation [21] is an important problem, we simplify the exposition by assuming a one-one relationship between semantic object classes and the nouns in the annotation.

the nouns n_s and n_p and the parameters of the appearance models(C_A) of these nouns. These parameters encode visual appearance of the object classes.

For every pair of image regions, there exist some relationships between them based on their locations and appearances. Relationship types are represented by a vocabulary of prepositions and comparative adjectives. Let r be a type of relationship (such as "above", "below") that holds between the objects associated with regions j and k. The nouns associated with the regions, n_s and n_p, provide priors on the types of relationships in which they might participate (For example, there is a high prior for the relationship "above" if the nouns are "sky" and "water", since in most images "sky" will occur above "water"). Every relationship is described by differential image features I_{jk}. The likelihood of the differential features depends on the type of relationship r and the parameters of the relationship model C_R.

4.2 Learning the Model

The training data consists of images annotated with nouns $(n_1^l, n_2^l..)$ and a set of relationships between these nouns represented by predicates $\mathcal{P}^l$, where l is the image number. Learning the model involves maximizing the likelihood of training images being associated with predicates given in the training data. The maximum likelihood parameters are the parameters of object and relationship classifiers, which are represented by $\theta = (C_A, C_R)$. However, evaluating the likelihood is expensive since it requires summation over all possible assignments of image regions to nouns. We instead treat the assignment as missing data and use an EM formulation to estimate θ^{ML}.

$$\theta^{ML} = \arg\max_{\theta} P(\mathcal{P}^1, \mathcal{P}^2..|I^1, I^2.., \theta) = \arg\max_{\theta} \sum_{A} P(\mathcal{P}^1, \mathcal{P}^2.., A|I^1, I^2.., \theta)$$

$$= \arg\max_{\theta} \prod_{l=1}^{N} \sum_{A^l} P(\mathcal{P}^l|I^l, \theta, A^l) P(A^l|I^l, \theta) \quad (1)$$

where A^l defines the assignment of image regions to annotated nouns in image l. Therefore, $A_i^l = j$ indicates that noun n_i^l is associated to region j in image l.

The first term in equation 1 represents the joint predicate likelihood given the assignments, classifier parameters and image regions. A predicate is represented as

Table 1. Notation

N: Number of images	l: Image under consideration (superscript)
$\mathcal{P}^l$: Set of Predicates for image l	$(n_1^l, n_2^l...)$: Set of Nouns for image l
$\mathcal{P}_i^l = r_i^l(n_{s_i}^l, n_{p_i}^l)$: i^{th} predicate	$A_i^l = j$: Noun n_i^l is associated with region j
C_A: Parameters of models of nouns	C_R:Parameters of models of relationships
r_i^l: Relationship represented by i^{th} predicate	
s_i: Index of noun which appears as argument1 in i^{th} predicate	
p_i: Index of noun which appears as argument2 in i^{th} predicate	
$I_{A_i^l}^l$: Image features for region assigned to noun n_i^l	

$r_i^l(n_{s_i}^l, n_{p_i}^l)$, where r_i^l is a relationship that exists between the nouns associated with region $A_{s_i}^l$ and $A_{p_i}^l$. We assume that each predicate is generated independently of others, given an image and assignment. Therefore, we rewrite the likelihood as:

$$\begin{aligned} P(\mathcal{P}^l|I^l,\theta,A^l) &= \prod_{i=1}^{|\mathcal{P}^l|} P(\mathcal{P}_i^l|I^l,A^l,\theta) \\ &\propto \prod_{i=1}^{|\mathcal{P}^l|} P(r_i^l|I^l_{A^l_{s_i}A^l_{p_i}},C_R)P(r_i^l|n_{s_i},n_{p_i}) \\ &\propto \prod_{i=1}^{|\mathcal{P}^l|} P(I^l_{A^l_{s_i}A^l_{p_i}}|r_i^l,C_R)P(r_i^l|C_R)P(r_i^l|n_{s_i},n_{p_i}) \end{aligned}$$

Given the assignments, the probability of associating a predicate $\mathcal{P}_i^l$ to the image is the probability of associating the relationship r_i^l to the differential features associated with the pair of regions assigned to n_{s_i} and n_{p_i}. Using Bayes rule, we transform this into the differential feature likelihood given the relationship word and the parameters of the classifier for that relationship word. $P(r_i^l|C_R)$ represents the prior on relationship words and is assumed uniform.

The second term in equation 1 evaluates the probability of an assignment of image regions to nouns given the image and the classifier parameters. Using Bayes rule, we rewrite this as:

$$\begin{aligned} P(A^l|I^l,\theta) &= \prod_{i=1}^{|A^l|} P(n_i^l|I^l_{A_i^l},C_A) \\ &\propto \prod_{i=1}^{|A^l|} P(I^l_{A_i^l}|n_i^l,C_A)P(n_i^l|C_A) \end{aligned}$$

where $|A^l|$ is the number of annotated nouns in the image, $P(I^l_{A_i^l}|n_i^l,C_A)$ is the image likelihood of the region assigned to the noun, given the noun and the parameters of the object model, $P(n_i^l|C_A)$ is the prior over nouns given the parameters of object models.

EM-Approach. We use an EM approach to simultaneously solve for the correspondence and for learning the parameters of classifiers represented by θ.

1. **E-step:** Compute the noun assignment for a given set of parameters from the previous iteration represented by θ^{old}. The probability of assignment in which noun i correspond to region j is given by:

$$P(A_i^l = j|\mathcal{P}^l,I^l,\theta^{old}) = \frac{\sum_{A'\in\mathcal{A}_{ij}^l} P(A'|\mathcal{P}^l,I^l,\theta^{old})}{\sum_k \sum_{A'\in\mathcal{A}_{ik}^l} P(A'|\mathcal{P}^l,I^l,\theta^{old})} \tag{2}$$

where $\mathcal{A}_{ij}$ refers to the subset of the set of all possible assignments for an image in which noun i is assigned to region j. The probability of any assignment A' for the image can be computed using Bayes rule:

$$P(A'|\mathcal{P}^l,I^l,\theta^{old}) \propto P(\mathcal{P}^l|A',I^l,\theta^{old})P(A'|I^l,\theta^{old}) \tag{3}$$

2. **M-step:** For the noun assignment computed in the E-step, we find the new ML parameters by learning both relationship and object classifiers. The ML parameters depend on the type of classifier used. For example, for a gaussian classifier we estimate the mean and variance for each object class and relationship class.

For initialization of the EM approach, we can use any image annotation approach with localization such as the translation based model described in [5]. Based on initial assignments, we initialize the parameters of both relationship and object classifiers.

We also want to learn the priors on relationship types given the nouns represented by $P(r|n_s, n_p)$. After learning the maximum likelihood parameters, we use the relationship classifier and the assignment to find possible relationships between all pairs of words. Using these generated relationship annotations we form a co-occurrence table which is used to compute $P(r|n_s, n_p)$.

4.3 Inference

Similar to training, we first divide the test image into regions. Each region j is associated with some features I_j and noun n_j. In this case, I_j acts as an observed variable and we have to estimate n_j. Previous approaches estimate nouns for regions independently of each other. We want to use priors on relationships between pair of nouns to constrain the labeling problem. Therefore, the assignment of labels cannot be done independently of each other. Searching the space of all possible assignments is infeasible.

We use a Bayesian network to represent our labeling problem and use belief propagation for inference. For each region, we have two nodes corresponding to the noun and image features from that region. For all possible pairs of regions, we have another two nodes representing a relationship word and differential features

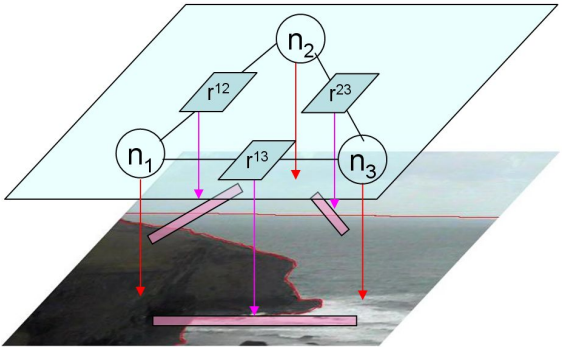


Fig. 3. An example of a Bayesian network with 3 regions. The r^{jk} represent the possible words for the relationship between regions (j, k). Due to the non-symmetric nature of relationships we consider both (j, k) and (k, j) pairs (in the figure only one is shown). The magenta blocks in the image represent differential features (I_{jk}).

from that pair of regions. Figure 3 shows an example of an image with three regions and its associated Bayesian network. The word likelihood is given by:

$$P(n_1, n_2..|I_1, I_2..I_{12}, .., C_A, C_R) \propto \prod_i P(I_i|n_i, C_A) \prod_{(j,k)} \sum_{r_{jk}} P(I_{jk}|r_{jk}, C_R) P(r_{jk}|n_j, n_k) \quad (4)$$

5 Experimental Results

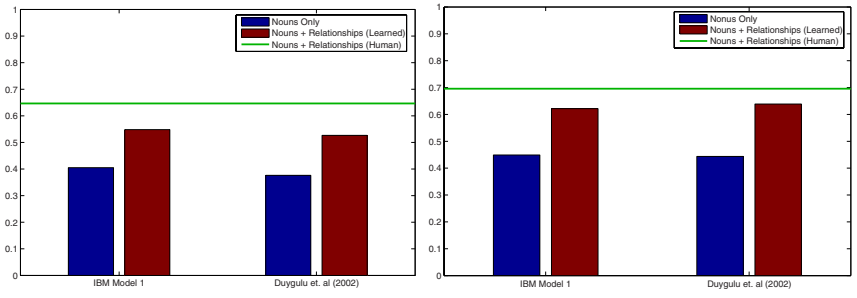
In all the experiments, we use a nearest neighbor based likelihood model for nouns and decision stump based likelihood model for relationships. We assume each relationship model is based on one differential feature(for example, the relationship "above" is based on difference in y locations of 2 regions). The parameter learning M-step therefore also involves feature selection for relationship classifiers. For evaluation we use a subset of the Corel5k training and test dataset used in [5]. For training we use 850 images with nouns and hand-labeled the relationships between subsets of pairs of those nouns. We use a vocabulary of 173 nouns and 19 relationships[4].

5.1 Resolution of Correspondence Ambiguities

We first evaluate the performance of our approach for the resolution of correspondence ambiguities in the training dataset. To evaluate the localization performance, we randomly sampled 150 images from the training dataset and compare it to human labeling. Similar to [22], we evaluate the performance in terms of two measures: "range of semantics identified" and "frequency correct". The first measure counts the number of words that are labeled properly by the algorithm. In this case, each word has similar importance regardless of the frequency with which it occurs. In the second case, a word which occurs more frequently is given higher importance. For example, suppose there are two algorithms one of which only labels 'car' properly and other which only labels 'sky' properly. Using the first measure, both algorithms have similar performance because they can correctly label one word each. However, using the second measure the latter algorithm is better as sky is more common and hence the number of correctly identified regions would be higher for the latter algorithm.

We compare our approach to image annotation algorithms which can be used for localization of nouns as well. These approaches are used to bootstrap our EM-algorithm. For our experiments, a co-occurrence based translation model [13] and translation based model with mixing probabilities [5] form the baseline algorithms. To show the importance of using "prepositions" and "comparative adjectives" for resolution of correspondence ambiguities, we use both algorithms to bootstrap EM and present our results. We also compare our performance with the algorithm where relationships are defined by a human instead of learning them from the dataset itself. Figure 4 compares the performance of all the

[4] Above, behind, below, beside, more textured, brighter, in, greener, larger, left, near, far from, ontopof, more blue, right, similar, smaller, taller, shorter.



(a) Semantic Range (b) Frequency Correct

Fig. 4. Comparison of normalized "semantic range" and "frequency correct" scores for the training dataset. The performance increases substantially by using prepositions and comparative adjectives in addition to nouns. The green line shows the performance when relationships are not learned but are defined by a human. The two red blocks show the performance of our approach where relationships and nouns are learned using the EM algorithm and bootstrapped by IBM Model1 or Duygulu et al. respectively.

algorithms with respect to the two measures described above. Figure 5 shows some examples of how ambiguity is removed using prepositions and comparative adjectives.

5.2 Labeling New Images

We also tested our model on labeling new test images. We used a subset of 500 test images provided in the Corel5k dataset. The subset was chosen based on the vocabulary of nouns learned from the training. The images were selected randomly from those images which had been annotated with the words present in our learned vocabulary. To find the missed labels we compute $\mathcal{S}_t \setminus \mathcal{S}_g$, where $\mathcal{S}_t$ is the set of annotations provided from the Corel dataset and $\mathcal{S}_g$ is the set of annotations generated by the algorithm. However, to test the correctness of labels generated by the algorithm we ask human observers to verify the annotations. We do not use the annotations in the Corel dataset since they contain only a subset of all possible nouns that describe an image. Using Corel annotations for evaluation can be misleading, for example, if there is "sky" in an image and an algorithm generates an annotation "sky" it may be labeled as incorrect because of the absence of sky from the Corel annotations. Figure 6 shows the performance of the algorithm on the test dataset. Using the proposed Bayesian model, the number of missed labels decreases by 24% for IBM Model 1 and by 17% for Duygulu et al. [5]. Also, using our approach 63% and 59% of false labels are removed respectively.

Figure 7 shows some examples of the labeling on the test set. The examples show how Bayesian reasoning leads to better labeling by applying priors on relationships between nouns. The recall and precision ratios for some common words in the vocabulary are shown in Table 2. The recall ratio of a word represents

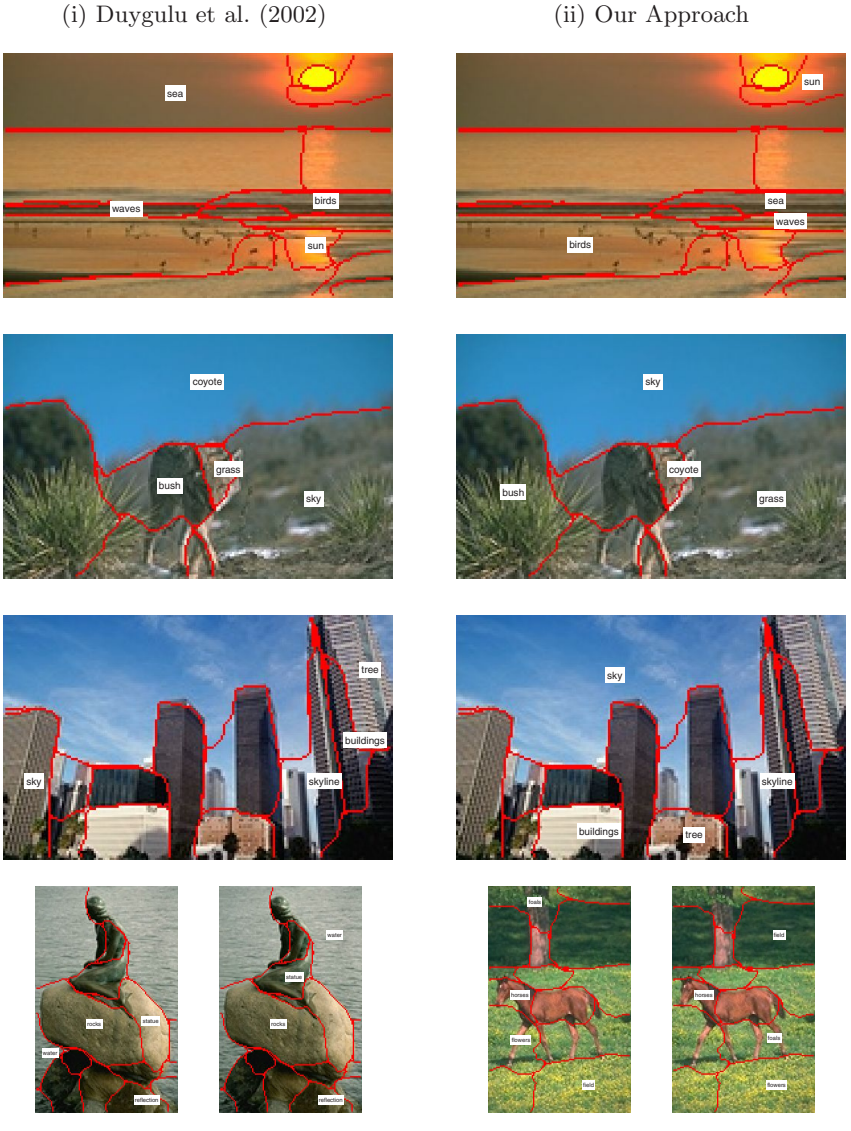
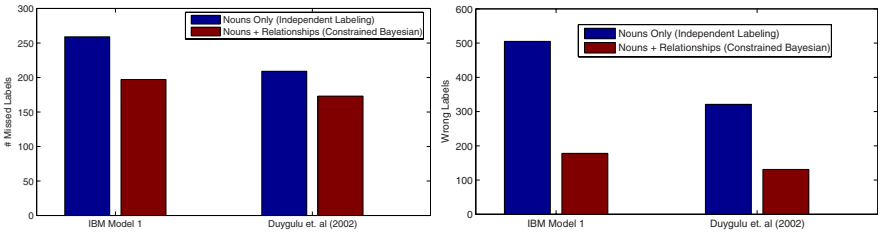


Fig. 5. Some examples of how correspondence ambiguity can be reduced using prepositions and comparative adjectives. Some of the annotations for the images are: **(a)** *near(birds,sea); below(birds,sun); above(sun, sea); larger(sea,sun); brighter(sun, sea); below(waves,sun)* **(b)** *below(coyote, sky); below(bush, sky); left(bush, coyote); greener(grass, coyote); below(grass,sky)* **(c)** *below(building, sky); below(tree,building); below(tree, skyline); behind(buildings,tree) blueish(sky, tree)* **(d)** *above(statue,rocks); ontopof(rocks, water); larger(water,statue)* **(e)** *below(flowers,horses); ontopof(horses, field); below(flowers,foals).*



(a) Missed Labels (b) False Labels

Fig. 6. Labeling performance on set of 100 test images. We do not consider localization errors in this evaluation. Each image has on average 4 labels in the Corel dataset.

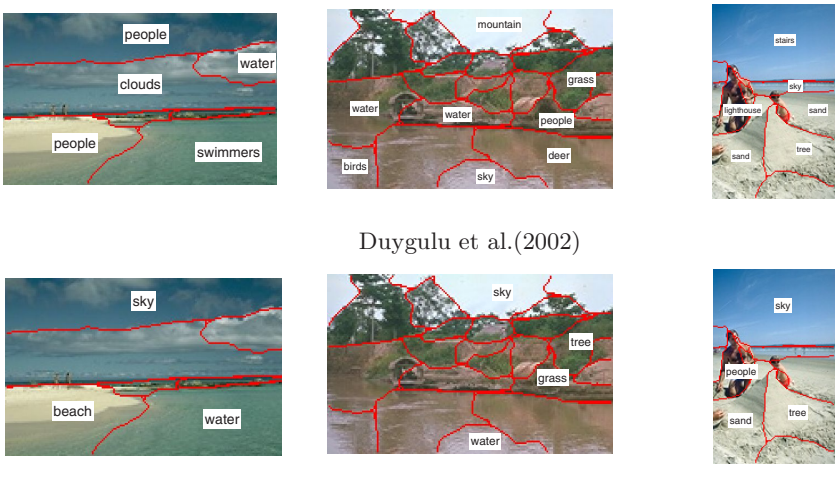


Fig. 7. Some examples of labeling on test dataset. By applying priors on relationships between different nouns, we can improve the labeling performance. For example, when labels are predicted independently, there can be labeling where region labeled "water" is above region labeled "clouds" as shown in the first image. This is however incongruent with the priors learned from training data where "clouds" are mostly above "water". Bayesian reasoning over such priors and likelihoods lead to better labeling performance.

the ratio of the number of images correctly annotated with that word using the algorithm to the number of images that should have been annotated with that word. The precision ratio of a word is the ratio of number of images that have been correctly annotated with that word to the number of images which were annotated with the word by the algorithm. While recall rates are reported with respect to corel annotations, precision rates are reported with respect to correctness defined by human observers. The results show that using a constrained bayesian model leads to improvement in labeling performance of common words in terms of both recall and precision rates.

Table 2. Precision-Recall Ratios

	water	grass	clouds	buildings	sun	sky	tree	
Recall	0.79	0.7	0.27	0.25	0.57	0.6	0.66	Duygulu(2002)
	0.90	1.0	0.27	0.42	0.57	0.93	0.75	Ours
Precision	0.57	0.84	0.76	0.68	0.77	0.98	0.70	Duygulu(2002)
	0.67	0.79	0.88	0.80	1.00	1.00	0.75	Ours

6 Conclusion

Learning visual classifiers from weakly labeled data is a hard problem which involves finding a correspondence between the nouns and image regions. While most approaches use a "bag" of nouns model and try to find correspondence using co-occurrence of image features and the nouns, correspondence ambiguity remains. We proposed the use of language constructs other than nouns, such as prepositions and comparative adjectives, to reduce correspondence ambiguity. While these relationships can be defined by humans, we present an EM based approach to simultaneously learn visual classifiers for nouns, prepositions and comparative adjectives. We also present a more constrained Bayesian model for the labeling process. Experimental results show that using relationship words helps in reduction of correspondence ambiguity and using a constrained model leads to a better labeling performance.

References

1. Armitage, L., Enser, P.: Analysis of user need in image archives. Journal of Information Science (1997)
2. Barnard, K., Duygulu, P., Freitas, N., Forsyth, D., Blei, D., Jordan, M.I.: Matching words and pictures. Journal of Machine Learning Research, 1107–1135 (2003)
3. Carneiro, G., Chan, A.B., Moreno, P., Vasconcelos, N.: Supervised learning of semantic classes for image annotation and retrieval. IEEE PAMI (2007)
4. Carbonetto, P., Freitas, N., Barnard, K.: A statistical model for general contextual object recognition. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 350–362. Springer, Heidelberg (2004)
5. Duygulu, P., Barnard, K., Freitas, N., Forsyth, D.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 97–112. Springer, Heidelberg (2002)
6. Barnard, K., Forsyth, D.: Learning the semantics of words and pictures. In: ICCV, pp. 408–415 (2001)
7. Andrews, S., Tsochantaridis, I., Hoffman, T.: Support vector machines for multiple-instance learning. In: NIPS (2002)
8. Li, J., Wang, J.: Automatic linguistic indexing of pictures by statistical modeling approach. IEEE PAMI (2003)
9. Maron, O., Ratan, A.: Multiple-instance learning for natural scene classification. ICML (1998)
10. Lavrenko, V., Manmatha, R., Jeon, J.: A model for learning the semantics of pictures. In: NIPS (2003)

11. Feng, S., Manmatha, R., Lavrenko, V.: Multiple bernoulli relevance models for image and video annotation. In: CVPR (2004)
12. Mori, Y., Takahashi, H., Oka, R.: Image to word transformation based on dividing and vector quantizing images with words. MISRM (1999)
13. Brown, P., Pietra, S., Pietra, V., Mercer, R.: The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics (1993)
14. Srikanth, M., Varner, J., Bowden, M., Moldovan, D.: Exploiting ontologies for automatic image annotation. SIGIR (2005)
15. Jin, R., Chai, J., Si, L.: Effective automatic image annotation via a coherent language model and active learning. Mutimedia (2004)
16. Brill, E.: A simple rule-based part of speech tagger. ACL (1992)
17. Brill, E.: Transformation-based error-driven learning and natural language processing. Computational Linguistics (1995)
18. Ferrari, V., Zisserman, A.: Learning visual attributes. In: NIPS (2007)
19. Barnard, K., Yanai, K., Johnson, M., Gabbur, P.: Cross modal disambiguation. Toward Category-Level Object Recognition (2006)
20. Barnard, K., Fan, Q.: Reducing correspondence ambiguity in loosely labeled training data. In: CVPR (2007)
21. Barnard, K., Johnson, M.: Word sense disambigutaion with pictures. AI (2005)
22. Barnard, K., Fan, Q., Swaminathan, R., Hoogs, A., Collins, R., Rondot, P., Kaufold, J.: Evaluation of localized semantics: data, methodology and experiments. Univ. of Arizona, TR-2005 (2005)

Learning Spatial Context: Using Stuff to Find Things

Geremy Heitz and Daphne Koller

Department of Computer Science, Stanford University
{gaheitz,koller}@cs.stanford.edu

Abstract. The sliding window approach of detecting rigid objects (such as cars) is predicated on the belief that the object can be identified from the appearance in a small region around the object. Other types of objects of amorphous spatial extent (e.g., trees, sky), however, are more naturally classified based on texture or color. In this paper, we seek to combine recognition of these two types of objects into a system that leverages "context" toward improving detection. In particular, we cluster image regions based on their ability to serve as context for the detection of objects. Rather than providing an explicit training set with region labels, our method automatically groups regions based on both their appearance and their relationships to the detections in the image. We show that our things and stuff (TAS) context model produces meaningful clusters that are readily interpretable, and helps improve our detection ability over state-of-the-art detectors. We also present a method for learning the active set of relationships for a particular dataset. We present results on object detection in images from the PASCAL VOC 2005/2006 datasets and on the task of overhead car detection in satellite images, demonstrating significant improvements over state-of-the-art detectors.

1 Introduction

Recognizing objects in an image requires combining many different signals from the raw image data. Figure 1 shows an example satellite image of a street scene, where we may want to identify all of the cars. From a human perspective, there are two primary signals that we leverage. The first is the local appearance in the window near the potential car. The second is our knowledge that cars appear on roads. This second signal is a form of contextual knowledge, and our goal in this paper is to capture this idea in a rigorous probabilistic model.

Recent papers have demonstrated that boosted object detectors can be effective at detecting monolithic object classes, such as cars [1] or faces [2]. Similarly, several works on multiclass segmentation have shown that regions in images can effectively be classified based on their color or texture properties [3]. These two lines of work have made significant progress on the problems of identifying "things" and "stuff," respectively. The important differentiation between these two classes of visual objects is summarized in Forsyth et al. [4] as:

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 30–43, 2008.

Fig. 1. (Left) An aerial photograph. (Center) Detected cars in the image (solid green = true detections, dashed red = false detections). (Right) Finding "stuff" such as buildings, by classifying regions, shown delineated by red boundaries.

Fig. 2. Example detections from the satellite dataset that demonstrate context. Classifying using local appearance only, we might think that both windows at left are cars. However, when seen in context, the bottom detection is unlikely to be an actual car.

> The distinction between materials — "stuff" — and objects — "things" — is particularly important. A material is defined by a homogeneous or repetitive pattern of fine-scale properties, but has no specific or distinctive spatial extent or shape. An object has a specific size and shape.

Recent work has also shown that classifiers for both things or stuff can benefit from the proper use of contextual cues. The use of context can be split into a number of categories. *Scene-Thing context* allows scene-level information, such as the scale or the "gist" [5], to determine location priors for objects. *Stuff-Stuff context* captures the notion that sky occurs above sea and road below building [6]. *Thing-Thing context* considers the co-occurrence of objects, encoding, for example, that a tennis racket is more likely to occur with a tennis ball than with a lemon [7]. Finally *Stuff-Thing context* allows the texture regions (e.g., the roads and buildings in Figure 1) to add predictive power to the detection of objects (e.g., the cars in Figure 1). We focus on this fourth type of context. Figure 2 shows an example of this context in the case of satellite imagery.

In this paper, we present a probabilistic model linking the detection of things to the unsupervised classification of stuff. Our method can be viewed as an attempt to cluster "stuff," represented by coherent image regions, into clusters that are both visually similar and best able to provide context for the detectable "things" in the image. Cluster labels for the image regions are probabilistically linked to the detection window labels through the use of region-detection "relationships," which encode their relative spatial locations. We call this model the things and stuff (TAS) context model because it links these two components

into a coherent whole. The graphical representation of the TAS model is depicted in Figure 3. At training time, our approach uses supervised (ground-truth) detection labels, without requiring supervised labels for the "stuff"-regions, and learns the model parameters using the Expectation-Maximization (EM) algorithm. Furthermore, we demonstrate a principled method for learning the set of active relationships (those used by the model). Our relationship selection through a variant of structural EM [8] is a novel aspect of our method, allowing the determination of which types of relationships are most appropriate without costly hand-engineering or cross-validation. At test time, these relationships are observed, and both the region labels and detection labels are inferred.

We present results of the TAS method on diverse datasets. Using the Pascal Visual Object Classes challenge datasets from 2005 and 2006, we utilize one of the top competitors as our baseline detector [9], and demonstrate that our TAS method improves the performance of detecting cars, bicycles and motorbikes in street scenes and cows and sheep in rural scenes (see Figure 5). In addition, we consider a very different dataset of satellite images from Google Earth, of the city and suburbs of Brussels, Belgium. The task here is to identify the cars from above; see Figure 2. For clarity, the satellite data is used as the running example throughout the paper, but all descriptions apply equally to the other datasets.

2 Related Work

The role of context in object recognition has become an important topic, due both to the psychological basis of context in the human visual system [10] and to the striking algorithmic improvements that "visual context" has provided [11].

The word "context" has been attached to many different ideas. One of the simplest forms is co-occurrence context. The work of Rabinovich et al. [7] demonstrates the use of this context, where the presence of a certain object class in an image probabilistically influences the presence of a second class. The context of Torralba et al. [11] assumes that certain objects occur more frequently in certain rooms, as monitors tend to occur in offices. While these methods achieve excellent results when many different object classes are labeled per image, they are unable to leverage unsupervised data for contextual object recognition.

In addition to co-occurrence context, many approaches take into account the spatial relationships between objects. At the descriptor level, Wolf et al. [12] detect objects using a descriptor with a large capture range, allowing the detection of the object to be influenced by surrounding image features. Because these methods use only the raw features, however, they cannot obtain a holistic view of an entire scene. This is analogous to addressing image segmentation using a wider feature window rather than a Markov random field (MRF). Similarly, Fink and Perona [13] use the output of boosted detectors for other classes as additional features for the detection of a given class. This allows the inclusion of signal beyond the raw features, but requires that all "parts" of a scene be supervised. Murphy et al. [5] use a global feature known as the "gist" to learn statistical priors on the locations of objects within the context of the specific

scene. The gist descriptor is excellent at predicting large structures in the scene, but cannot handle the local interactions present in the satellite data, for example.

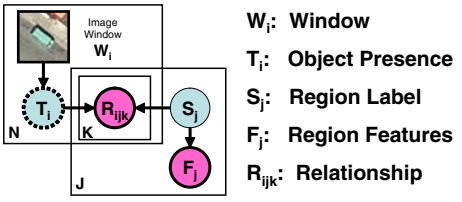
Another approach to modeling spatial relationships is to use a Markov Random Field (MRF) or variant (CRF,DRF) [14, 15] to encode the preferences for certain spatial relationships. These techniques offer a great deal of flexibility in the formulation of the affinity function and all the standard benefits of a graphical model formulation (e.g., well-known learning and inference techniques). Singhal et al. [6] also use similar concepts to the MRF formulation to aggregate decisions across the image. These methods, however, suffer from two drawbacks. First, they tend to require a large amount of annotation in the training set. Second, they put things and stuff on the same footing, representing both as "sites" in the MRF. Our method requires less annotation and allows detections and image regions to be represented in their (different) natural spaces.

Perhaps the most ambitious attempts to use context involves the attempt to model the scene of an image holistically. Torralba [1], for instance, uses global image features to "prime" the detector with the likely presence/absence of objects, the likely locations, and the likely scales. The work of Hoiem and Efros [16] takes this one level further by explicitly modeling the 3D layout of the scene. This allows the natural use of scale and location constraints (e.g., things closer to the camera are larger). Their approach, however, is tailored to street scenes, and requires domain-specific modeling. The specific form of their priors would be useless in the case of satellite images, for example.

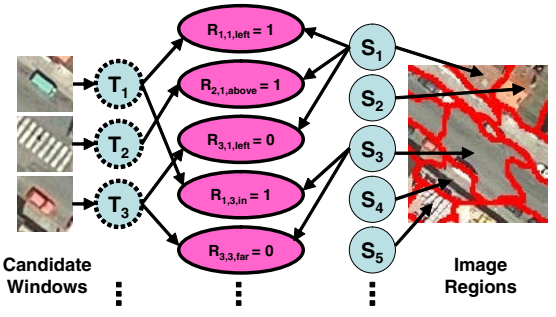
3 Things and Stuff (TAS) Context Model

Our probabilistic context model builds on two standard components that are commonly used in the literature. The first is sliding window object detection, and the second is unsupervised image region clustering. A common method for finding "things" is to slide a window over the image, score each window's match to the model, and return the highest matching such windows. We denote the features in the i^{th} candidate window by W_i, the presence/absence of the target class in that window by T_i (T for "thing"), and assume that what we learn in our detector is a conditional probability $P(T_i \mid W_i)$ from the window features to the probability that the window contains the object; this probability can be derived from most standard classifiers, such as the highly successful boosting approaches [1]. The set of windows included in the model can, in principle, include all windows in the image. We, however, limit ourselves to looking at the top scoring detection windows according to the detector (i.e., all windows above some low threshold, chosen in order to capture most of the true positives).

The second component we build on involves clustering coherent regions of the image into groups based on appearance. We begin by segmenting the image into regions, known as superpixels, using the normalized cut algorithm of Ren and Malik [17]. For each region j, we extract a feature vector $\boldsymbol{F}_j$ that includes color and texture features. For our stuff model, we will use a generative model where each region has a hidden class, and the features are generated from a Gaussian



(a) TAS plate model



(b) TAS ground network

Fig. 3. The TAS model. The plate representation (a) gives a compact visualization of the model, which unrolls into a "ground" network (b) for any particular image. Nodes with dotted lines represent variables that are unobserved during training; only the pink nodes (which represent image features) are observed during testing.

distribution with parameters depending on the class. Denote by S_j (S for "stuff") the (hidden) class label of the j^{th} region. We assume that the features are derived from a standard naive Bayes model, where we have a probability distribution $P(\boldsymbol{F}_j \mid S_j)$ of the image features given the class label.

In order to relate the detector for "things" (T's) to the clustering model over "stuff" (S's), we need to develop an intuitive representation for the relationships between these units. Human knowledge in this area comes in sentences like "cars drive on roads," or "cars park 20 feet from buildings." We introduce indicator variables R_{ijk} that indicate whether candidate detection i and region j have relationship k. The different k's represent different relationships, such as: "detection i is *in* region j" ($k = 1$), or "detection i is about 100 pixels away from region j" ($k = 2$). For now we assume the set of relationships (the meaning of R_{ijk} for each k) is known, and in Section 4 we describe how this set of relationships is learned from the data.

We can now link our component models into a single coherent probabilistic things and stuff (TAS) model, as depicted in the plate model of Figure 3(a). Probabilistic influence flows between the detection window labels and the image region labels through the v-structures that are activated by observing the relationship variables. For a particular input image, this plate model unrolls into a "ground" network that encodes a distribution over the detections and regions of the image. Figure 3(b) shows a toy example of a partial ground network for the

image of Figure 1. It is interesting to note the similarities between TAS and the MRF approaches in the literature. In effect, the relationship variables link the detections and the regions into a probabilistic web where signals at one point in the web influence a detection in another part of the image, which in turn influences the region label in yet another location. Although similar to an MRF in its ability to define a holistic view of the entire image, our model is generative, allowing us to train very efficiently, even when the S_j variables are unobserved.

All variables in the TAS model are discrete except for the feature variables $\boldsymbol{F}_j$. This allows for simple table conditional probability distributions (CPDs) for all discrete nodes in this Bayesian network. The probability distribution over these variables decomposes according to:

$$P(\boldsymbol{T}, \boldsymbol{S}, \boldsymbol{F}, \boldsymbol{R} \mid \boldsymbol{W}) = \prod_i P(T_i \mid W_i) \prod_j P(S_j) P(\boldsymbol{F}_j \mid S_j) \prod_{ijk} P(R_{ijk} \mid T_i, S_j).$$

We note that TAS is very modular. It allows us to "plug in" any detector and any generative model for regions (e.g., [3]).

4 Learning and Inference in the TAS Model

Because TAS unrolls into a Bayesian network for each image, we can use standard learning and inference methods. In particular, we learn the parameters of our model using the Expectation-Maximization (EM) [18] algorithm and perform inference using Gibbs sampling, a standard variant of MCMC sampling [19]. Furthermore, we show how to learn the set of active relationships from a large candidate relationship pool using a structure search interleaved with the EM [8].

Learning the Parameters with EM. At learning time, we have a set of images with annotated labels for our target object class(es). We first train the base detectors using this set, and select as our candidate windows all detections above a threshold; we thus obtain a set of candidate detection windows $W_1 \ldots W_N$ along with their ground-truth labels $T_1 \ldots T_N$. We also have a set of regions and a feature vector F_j for each, and an R_{ijk} relationship variable for every window-region pair and relationship type, which is observed for each pair based on their relative spatial relationship. Our goal is to learn parameters for the TAS model.

Because our variables are discrete and have relatively small cardinality, we can learn our model using the EM algorithm. EM iterates between using probabilistic inference to derive a soft completion of the hidden variables (E-step) and finding maximum-likelihood parameters relative to this soft completion (M-step). The E-step here is particularly easy: at training time, only the S_j's are unobserved; moreover, because the T variables are observed, the S_j's are conditionally independent of each other. Thus, the inference step turns into a simple computation for each S_j separately, a process which can be performed in linear time. The M-step for table CPDs can be performed easily in closed form. To provide a good starting point for EM, we initialize the cluster assignments using the K-means algorithm. EM is guaranteed to converge to a local maximum of the likelihood function of the observed data.

Learning the Relationships. So far, we have assumed a known set of relationships. However, because different data may require different types of contextual relationships, we would like to learn which to use. We begin by defining a large set $\mathcal{C}$ of "candidate relationships" (i.e., all possible relationships that we want to consider for inclusion). For instance, we may include both "above by 100 pixels," and "above by 200 pixels" in $\mathcal{C}$, even though we believe only one of these will be chosen. Our goal is to search through $\mathcal{C}$ for the subset of relationships that will best facilitate the use of context. We denote this "active" set by $\mathcal{R}$.

Intuitively, if a particular type of relationship (e.g., , "above by 100 pixels") is "inactive" then we want to force the value of the corresponding R_{ijk} variables to be independent of the value of the T_i's and S_j's. In the language of Bayesian networks, we can achieve this by removing the edges from all T_i and S_j variables to the R_{ijk} variables for this particular k. With this view of "activating" relationships by including the edges in the Bayesian Network, we can formulate our search for $\mathcal{R}$ as a structure learning problem. To learn this network structure, we turn to the method of structural EM [8]. In particular, if we are considering K possible relationships, there are 2^K possible subsets to consider (each relationship can be "active" or "inactive"). We search this space using a greedy hill-climbing approach that is interleaved with the EM parameter learning. The hill-climbing begins with an empty set of relationships ($\mathcal{R} = \emptyset$), and adds or removes relationships one at a time until a local maximum is reached.

Standard structural EM scores each network using the log probability of the expected data, which is easily computed from the output of the E-step above. However, because our final goal is to correctly classify the "things," we would rather score each structure using the log probability of the T_i's. While this requires us to perform inference (described below) for each candidate structure, it is both theoretically more sound and produced far better results than the standard score. In order to avoid overfitting, we initially used a BIC penalty, but found that this resulted in too few "active" relationships. Instead, in experiments below, we include a Gaussian prior over the number of active relationships. Our

Algorithm `LearnTAS`
Input: Candidate relationships $\mathcal{C}$, Dataset $\mathcal{D} = \{(\boldsymbol{W}[m], \boldsymbol{T}[m], \boldsymbol{F}[m], \boldsymbol{R}[m])\}$
$\mathcal{R} \leftarrow \emptyset$ (all relationships "inactive")
Repeat until convergence
Repeat until convergence (EM over Parameters)
E-step: $Q[m] \leftarrow P(\boldsymbol{S} \mid \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R}; \theta_{\mathcal{R}}) \quad \forall m$
M-step: $\theta_{\mathcal{R}} \leftarrow \text{argmax} E_Q \left[\sum_m \ell(\boldsymbol{S}, \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R} \mid \boldsymbol{W}; \theta_{\mathcal{R}})\right]$
Repeat until convergence (Greedy Structure Search)
Forall k, $score_k = \sum_m \ell(\boldsymbol{T} \mid \boldsymbol{F}, \boldsymbol{R}; \theta_{\mathcal{R} \oplus k})$ (score with k "activated")
$\mathcal{R} \leftarrow \mathcal{R} \oplus k^*$ where $k^* = \text{argmax } score_k$
Return Set $\mathcal{R}$ of "active" relationships, TAS parameters $\theta_{\mathcal{R}}$

Fig. 4. Learning a TAS model. Here ℓ represents the log-likelihood of the data, and $\oplus$ represents the set exclusive-or operation.

learning process outputs an active set of relationships, $\mathcal{R}$, and the parameters of the TAS model for that set, $\theta_{\mathcal{R}}$. The algorithm is outlined in Figure 4.

Inference with Gibbs Sampling. At test time, our system must determine which windows in a new image contain the target object. We observe the candidate detection windows (W_i's, extracted by thresholding the base detector output), the features of each image region ($\boldsymbol{F}_j$'s), and the relationships (R_{ijk}'s). Our task is to find the probability that each window contains the object:

$$P(\boldsymbol{T} \mid \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{W}) = \sum_{\boldsymbol{S}} P(\boldsymbol{T}, \boldsymbol{S} \mid \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{W}) \tag{1}$$

Unfortunately, this expression involves a summation over an exponential set of values for the $\boldsymbol{S}$ vector of variables. We solve the inference problem approximately using a Gibbs sampling [19] MCMC method. We begin with some assignment to the variables. Then, in each Gibbs iteration we first resample all of the S's and then resample all the T's according to the following two probabilities:

$$P(S_j \mid \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{W}) \propto P(S_j)P(F_j \mid S_j) \prod_{ik} P(R_{ijk} \mid T_i, S_j) \tag{2}$$

$$P(T_i \mid \boldsymbol{S}, \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{W}) \propto P(T_i \mid W_i) \prod_{jk} P(R_{ijk} \mid T_i, S_j). \tag{3}$$

These sampling steps can be performed efficiently, as the T_i variables are conditionally independent given the S's and the S_j's are conditionally independent given the T's. In the last Gibbs iteration for each sample, rather than resampling $\boldsymbol{T}$, we compute the posterior probability over $\boldsymbol{T}$ given our current $\boldsymbol{S}$ samples, and use these distributional particles for our estimate of the probability in (1).

5 Experimental Results

In order to evaluate the TAS model, we perform experiments on three datasets. The first two are from the PASCAL Visual Object Classes challenges 2005 and 2006[20]. The scenes are urban and rural, indoor and outdoor, and there is a great deal of scale variation amongst the objects. The third is a set of satellite images acquired from Google Earth, with the goal of detecting cars. Because of the impoverished visual information, there are many false positives when a sliding window detector is applied. In this case, context provides a filtering mechanism to remove the false positives. Because these two applications are different, we use different detectors for each. We allow S a cardinality of $|S| = 10$,[1] and use 44 features for the image regions (color, texture, shape, etc. [21]).

PASCAL VOC Datasets. For these experiments, we used four classes from the VOC2005 data, and two classes from the VOC2006 data. The VOC2005 dataset consists of 2232 images, manually annotated with bounding boxes for

[1] Results were robust to a range of $|S|$ between 5 and 20.

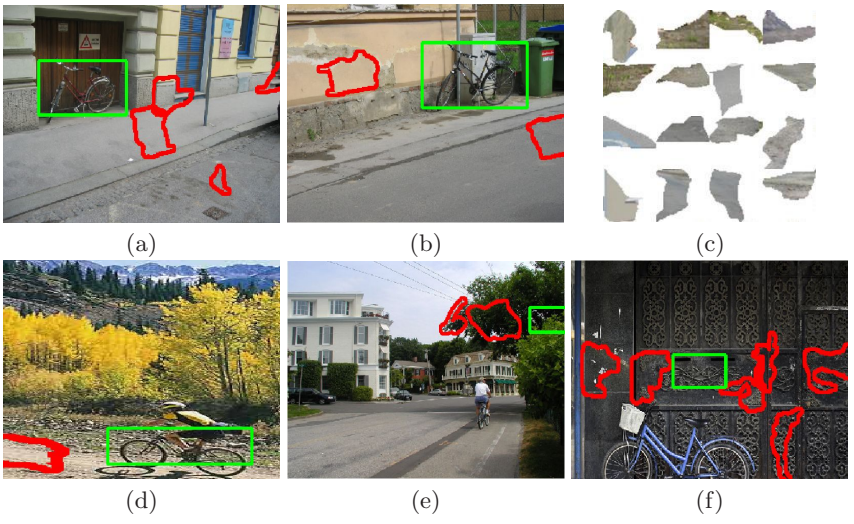
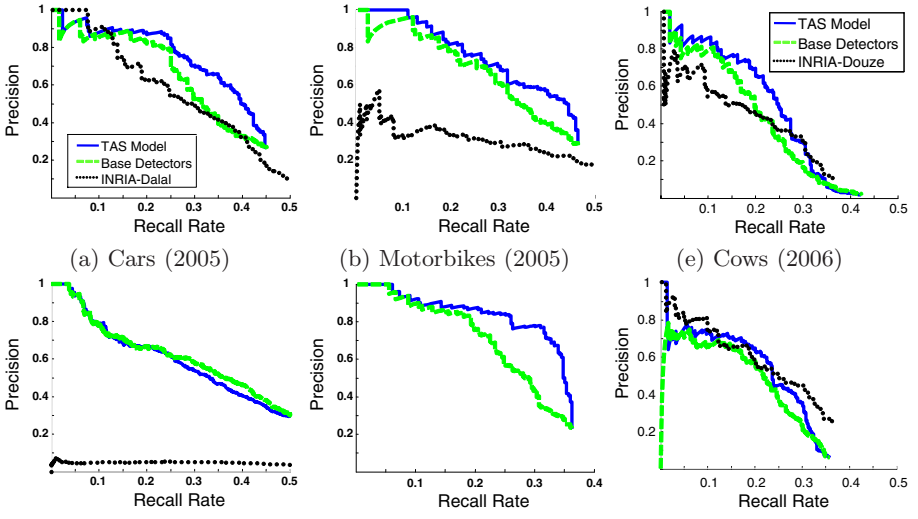


Fig. 5. (a,b) Example training detections from the bicycle class, with detection windows outlined by the green rectangles. The image regions with active relationships to the detection window are outlined in red. (c) 16 of the most representative regions for cluster #3. This cluster corresponds to "roads" or "bushes" as things that are gray/green and occur near cars. (d) A case where context helped find a true detection. (e,f) Two examples where incorrect detections are filtered out by context.

four image classes: cars, people, motorbikes, and bicycles. We use the "train+val" set (684 images) for training, and the "test2" set (859 images) for testing. The VOC2006 dataset contains 5304 images, manually annotated with 12 classes, of which we use the cow and sheep classes. We train on the "trainval" set (2618 images) and test on the "test" set (2686 images). To compare with the results of the challenges, we adopted as our detector the HOG (histogram of oriented gradients) detector of Dalal and Triggs [9]. This detector uses an SVM and therefore outputs a score $\text{margin}_i \in (-\infty, +\infty)$, which we convert into a probability by learning a logistic regression function for $P(T_i \mid \text{margin}_i)$. We also plot the precision-recall curve using the code provided in the challenge toolkit.

We learned our model with a set of 25 candidate relationships that included regions within the window, at offsets in eight directions (every 45 degrees) at two different distances, and the union of various combinations of features (e.g., R_{24} indicates regions to the right *or* left of the window by one bounding box). Figure 5 (top row) shows example bicycle detection candidates, and the related image regions, suggesting the type of context that might be learned. For example, the region beside and below both detections (outlined in red) belongs to cluster #3, which looks visually like a road or bush cluster (see Figure 5(c)). The learned values of the model parameters also indicate that being to the left or right of this cluster increases the probability of a window containing a bicycle (e.g., by about 33% in the case where $R_{ijk} = 1$ for this relationship).



(a) Cars (2005) (b) Motorbikes (2005) (e) Cows (2006)

(c) People (2005) (d) Bicycles (2005) (f) Sheep (2006)

Object Class	Base AP	TAS AP (Fixed R)	TAS AP (Learned R)	Improvement (TAS - Base)
Cars	0.325	0.360	**0.363**	0.038
Motorbikes	0.341	**0.390**	0.373	0.032
People	**0.346**	0.346	0.337	-0.009
Bicycles	0.281	0.310	**0.325**	0.044
Cows	0.224	0.241	**0.258**	0.034
Sheep	0.206	0.233	**0.248**	0.042

Fig. 6. (top) Precision-recall (PR) curves for the VOC classes. (bottom) Average precision (AP) scores for each experiment. AP is a robust variant of the area under the PR curve. We show the AP score for TAS with hand-selected relationships as well as with learned relationships, with results from the best performing model in bold.

We performed a single run of EM learning with structure search to convergence, which takes 1-2 hours on an Intel Dual Core 1.9 GHz machine with 2 GB of memory. We run separate experiments for each class, though in principle it would be possible to learn a single joint model over all classes. By separating the classes, we are able to isolate the contextual contribution from the stuff, rather than between the different types of things present in the images. For our MCMC inference, we found that, due to the strength of the baseline detectors, the Markov chain converged fairly rapidly; we achieved very good results using merely 10 MCMC samples, where each is initialized randomly and then undergoes 5 Gibbs iterations. Inference takes about 0.5 seconds per image.

The bottom row of Figure 5 shows some detections that were corrected using context. We show one example where a true bicycle was discovered using context, and two examples where false positives were filtered out by our model. These examples demonstrate the type of information that is being leveraged by TAS.

In the first example, the dirt road to the left of the window gives a signal that this detection is at ground level, and is therefore likely to be a bicycle.

Figure 6 shows the full recall-precision curve for each class. For (a-d) we compare to the 2005 **INRIA-Dalal** challenge entry, and for (e,f) we compare to the 2006 **INRIA-Douze** entry, both of which used the HOG detector. We also show the curve produced by our **Base Detector** alone.[2] Finally, we plot the curves produced by our **TAS Model**, trained using full EM, which scores windows using the probability of (1). From these curves, we see that the TAS model provided an improvement in accuracy for all but the "people" class. We believe the lack of improvement for people is due to the wide variation of backgrounds in these images, providing no strong context cues to latch onto. Furthermore, the base HOG detector was in fact originally optimized to detect people.

Satellite Images. The second dataset is a set of 30 images extracted from Google Earth. The images are color, and of size 792×636, and contain 1319 manually labeled cars. The average car window is approximately 45×45 pixels, and all windows are scaled to these dimensions for training. We used 5-fold cross-validation, and results below report the mean performance across the folds.

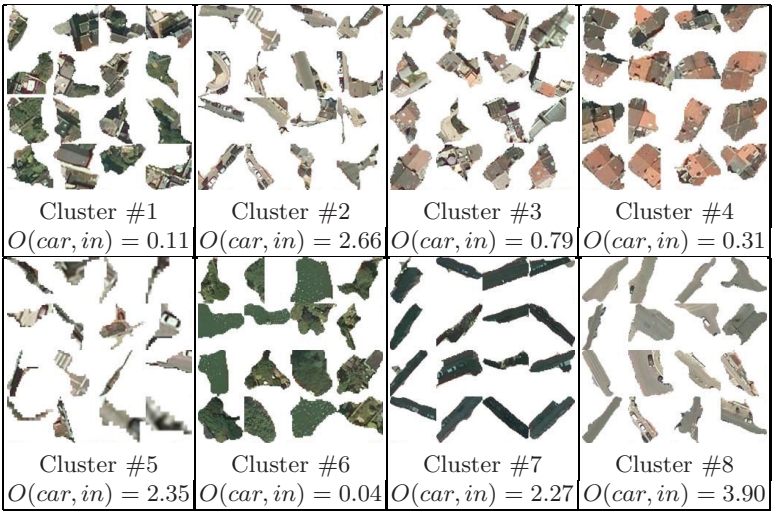
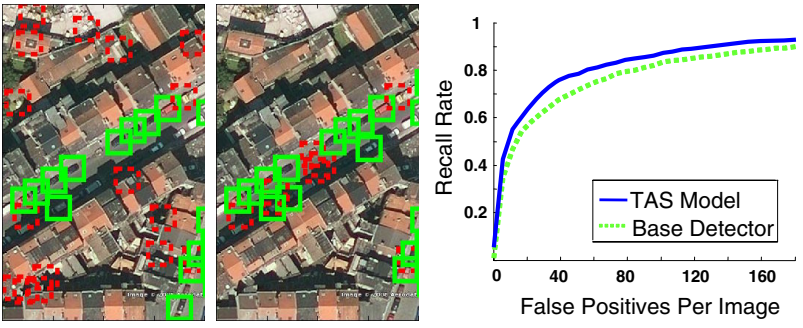


Fig. 7. Example clusters learned by the context model on the satellite dataset. Each cluster shows 16 of the training image regions that are most likely to be in the cluster based on $P(\boldsymbol{F} \mid S)$. For each cluster, we also show the odds-ratio of a window "in" a region labeled by that cluster containing a car ($O(car, in) = P(in|car, q)/P(in|no\ car, q)$). A higher odds-ratio indicates that this contextual relationship increases the model's confidence that the window contains a car.

[2] Differences in PR curves between our base detector and the **INRIA-Dalal**/**INRIA-Douze** results come from the use of slightly different training windows and parameters. **INRIA-Dalal** did not report results for "bicycle."



(a) Base Detectors (b) TAS Detections (c) PR Curve

Fig. 8. Example image, with detections found by the base detector (a), and by the TAS model (b) with a threshold of 0.15. The TAS model filters out many of the false positives far away from roads. (c) shows a plot of recall rate vs. false positives per image for the satellite data. The results here are averaged across 5 folds, and show a significant improvement from using TAS over the base detectors.

Here, we use a patch-based boosted detector very similar to that of Torralba [1]. We use 50 rounds of boosting with two level decision trees over patch cross-correlation features that were computed for 15,000–20,000 rectangular patches (intensity and gradient) of various aspect ratios and widths of 4-22 pixel. As above, we convert the boosting score into a probability using logistic regression. For training the TAS model, we used 10 random restarts of EM, selecting the parameters that provided the best likelihood of the observed data. For inference, we need to account for the fact that our detectors are much weaker, and so more samples are necessary to adequately capture the posterior. We utilize 20 samples per image, where each sample undergoes 20 iterations.

Figure 7 shows some learned "stuff" clusters. Eight of the ten learned clusters are shown, visualized by presenting 16 of the image regions that rank highest with respect to $P(\boldsymbol{F} \mid S)$. These clusters have a clear interpretation: cluster #4, for instance, represents the roofs of houses and cluster #6 trees and water regions. With each cluster, we also show the odds-ratio of a candidate window containing a car given that it is in this region. Clusters #7 and #8 are road clusters, and increase the chance of a nearby window being a car by a factor of 2 or more. Clusters #1 and #6, however, which represent forest and grass areas, decrease the probability of nearby candidates being cars by factors of 9 or more. Figure 8 shows an example with the detections of the detector alone and of the TAS model, which filters out many of the false positives that are not near roads. Because there are many detections per image, we plot the recall versus the number of false detections per image in Figure 8(c). The **Base Detectors** are compared to the **TAS Model**, verifying that context indeed improves our results, by filtering out many of the false positives.

6 Discussion and Future Directions

In this paper, we have presented the TAS model, a probabilistic framework that captures the contextual information between "stuff" and "things", by linking discriminative detection of objects with unsupervised clustering of image regions. Importantly, the method does not require extensive labeling of image regions; standard labeling of object bounding boxes suffices for learning a model of the appearance of stuff regions and their contextual cues. We have demonstrated that the TAS model improves the performance even of strong base classifiers, including one of the top performing detectors in the PASCAL challenge.

The flexibility of the TAS model provides several important benefits. The model can accommodate almost any choice of object detector that produces a score for candidate windows. It is also flexible to any generative model over any type of region features. For instance, we might pre-cluster the regions into visual words, and then use a multinomial distribution over these words [21]. Additionally, because our model discovers which relationships to use, our method has the ability to discover spatial interactions that are not already known to the modeler. Indeed, automated structure-learning such as the one we employ here can provide a valuable substitute for the laborious process of manual feature construction that is necessary for engineering a computer vision system.

Because the image region clusters are learned in an unsupervised fashion, they are able to capture a wide range of possible concepts. While a human might label the regions in one way (say trees and buildings), the automatic learning procedure might find a more contextually relevant grouping. For instance, the TAS model might split buildings into two categories: apartments, which often have cars parked near them, and factories, which rarely co-occur with cars.

As discussed in Section 2, recent work has amply demonstrated the importance of context in computer vision. The context modeled by the TAS framework is a natural complement for many of the other types of context in the literature. In particular, while many other forms of context can relate known objects that have been labeled in the data, our model can extract the signals present in the unlabeled part of the data. However, a major limitation of the TAS model is that it captures only 2D context. This issue also affects our ability to determine the appropriate scale for the contextual relationships. It would be interesting to integrate a TAS-like definition of context into an approach that attempts some level of 3D reconstruction, such as the work of Hoiem and Efros [16] or of Saxena et al. [22], allowing us to utilize 3D context and address the issue of scale.

References

[1] Torralba, A.: Contextual priming for object detection. IJCV 53(2) (2003)
[2] Viola, P., Jones, M.: Robust real-time face detection. In: ICCV (2001)
[3] Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 1–15. Springer, Heidelberg (2006)

[4] Forsyth, D.A., Malik, J., Fleck, M.M., Greenspan, H., Leung, T.K., Belongie, S., Carson, C., Bregler, C.: Finding pictures of objects in large collections of images. In: Object Representation in Computer Vision (1996)
[5] Murphy, K., Torralba, A., Freeman, W.: Using the forest to see the tree: a graphical model relating features, objects and the scenes. In: NIPS (2003)
[6] Singhal, A., Luo, J., Zhu, W.: Probabilistic spatial context models for scene content understanding. In: CVPR (2003)
[7] Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in context. In: ICCV (2007)
[8] Friedman, N.: Learning belief networks in the presence of missing values and hidden variables. In: ICML (1997)
[9] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
[10] Oliva, A., Torralba, A.: The role of context in object recognition. Trends Cogn. Sci. (2007)
[11] Torralba, A., Murphy, K., Freeman, W., Rubin, M.: Context-based vision system for place and object recognition. In: ICCV (2003)
[12] Wolf, L., Bileschi, S.: A critical view of context. IJCV 69(2) (2006)
[13] Fink, M., Perona, P.: Mutual boosting for contextual inference. In: NIPS (2003)
[14] Kumar, S., Hebert, M.: A hierarchical field framework for unified context-based classification. In: ICCV (2005)
[15] Carbonetto, P., de Freitas, N., Barnard, K.: A statistical model for general contextual object recognition. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 350–362. Springer, Heidelberg (2004)
[16] Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. In: CVPR (2006)
[17] Ren, X., Malik, J.: Learning a classification model for segmentation. In: ICCV (2003)
[18] Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39(1) (1977)
[19] Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions, and the bayesian restoration of images (1987)
[20] Everingham, M.: The 2005 pascal visual object classes challenge. In: MLCW (2005)
[21] Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D.M., Jordan, M.I.: Matching words and pictures. JMLR 3 (2003)
[22] Saxena, A., Sun, M., Ng, A.Y.: Learning 3-d scene structure from a single still image. In: CVPR (2007)

Segmentation and Recognition Using Structure from Motion Point Clouds

Gabriel J. Brostow[1], Jamie Shotton[2], Julien Fauqueur[3], and Roberto Cipolla[4]

[1] University College London and ETH Zurich
[2] Microsoft Research Cambridge
[3] University of Cambridge (now with MirriAd Ltd.)
[4] University of Cambridge

Abstract. We propose an algorithm for semantic segmentation based on 3D point clouds derived from ego-motion. We motivate five simple cues designed to model specific patterns of motion and 3D world structure that vary with object category. We introduce features that project the 3D cues back to the 2D image plane while modeling spatial layout and context. A randomized decision forest combines many such features to achieve a coherent 2D segmentation and recognize the object categories present. Our main contribution is to show how semantic segmentation is possible based *solely on motion-derived 3D world structure*. Our method works well on sparse, noisy point clouds, and unlike existing approaches, does not need appearance-based descriptors.

Experiments were performed on a challenging new video database containing sequences filmed from a moving car in daylight and at dusk. The results confirm that indeed, accurate segmentation and recognition are possible using only motion and 3D world structure. Further, we show that the motion-derived information complements an existing state-of-the-art appearance-based method, improving both qualitative and quantitative performance.

1 Introduction

We address the question of whether motion and 3D world structure can be used to accurately segment video frames and recognize the object categories present. In particular, as illustrated in Fig. 1, we investigate how to perform semantic segmentation from the sparse, noisy 3D point cloud given by structure from ego-motion. Our algorithm is able to accurately recognize objects and segment video frames without appearance-based descriptors or dense depth estimates obtained using *e.g.,* dense stereo or laser range finders. The structure from motion, or SfM, community [1] has demonstrated the value of ego-motion derived data, and their modeling efforts have even extend to stationary geometry of cities [2]. However, the *object recognition* opportunities presented by the inferred motion and structure have largely been ignored[1].

[1] The work of [3] was similarly motivated, and used laser-scans of static scenes to compute a 3D planar patch feature, which helped to train a chain of binary classifiers.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 44–57, 2008.

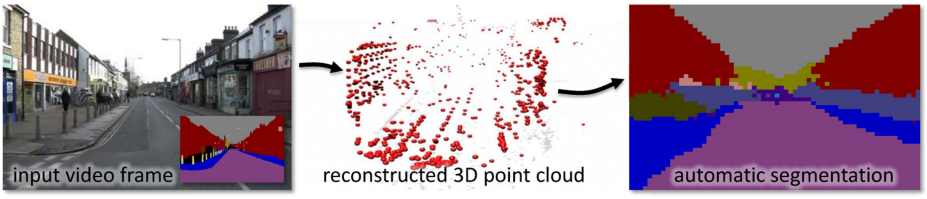


Fig. 1. The proposed algorithm uses 3D point clouds estimated from videos such as the pictured driving sequence (with ground truth inset). Having trained on point clouds from other driving sequences, our new motion and structure features, based purely on the point cloud, perform 11-class semantic segmentation of each test frame. The colors in the ground truth and inferred segmentation indicate category labels.

The proposed algorithm uses camera-pose estimation from video as an existing component, and assumes ego-motion is the dominant cause of pixel flow [4]. Tracked 2D image features are triangulated to find their position in world space and their relationship to the moving camera path. We suggest five simple motion and structure cues that are indicative of object categories present in the scene. Projecting these cues from the 3D point cloud to the 2D image, we build a randomized decision forest classifier to perform a coherent semantic segmentation.

Our main contributions are: (i) a demonstration that semantic segmentation is possible based *solely on motion-derived 3D world structure*; (ii) five intuitive motion and structure cues and a mechanism for projecting these 3D cues to the 2D image plane for semantic segmentation; and (iii) a challenging new database of video sequences filmed from a moving car and hand-labeled with ground-truth semantic segmentations. Our evaluation shows performance comparable to existing state-of-the-art appearance based techniques, and further, that our motion-derived features complement appearance-based features, improving both qualitative and quantitative performance.

Background. An accurate automatic scene understanding of images and videos has been an enduring goal of computer vision, with applications varying from image search to driving safety. Many successful techniques for 2D object recognition have used individual still images [5,6,7]. Without using SfM, Hoiem et al. [8,9] achieve exciting results by considering several spatial cues found in single images, such as surface orientations and vanishing points, to infer the camera viewpoint or general scene structure. This, in turn, helps object recognition algorithms refine their hypotheses, culling spatially infeasible detections. 3D object recognition is still a new research area. Huber et al.[10] matched laser rangefinder data to learned object models. Other techniques build 3D object models and match them to still images using local descriptors [11,12,13,14]. None of these methods, however, can exploit the motion-based cues available in video sequences. Dalal et al. [15] is a notable exception that used differential optical flow in pairs of images. In this paper, we reason about the moving 3D scene given a moving 2D camera. Our method works well on sparse, noisy point clouds, and does not need appearance-based descriptors attached to 3D world points.

There is a long history of fascinating research about motion-based recognition of human activities [16]. Laptev and Lindeberg [17] introduced the notion of space-time interest points to help detect and represent sudden actions as high gradient points in the *xyt* cube for motion-based activity recognition. Our focus is rather object recognition, and our features do not require a stationary camera.

While it is tempting to apply other detectors (*e.g.,* pedestrians [18]) directly to the problem of recognizing objects from a moving camera, motion compensation and motion segmentation are still relatively open problems. Yin et al. [19] use low-level motion cues for bi-layer video segmentation, though do not achieve a semantic labeling. Computer vision for driving has proven challenging and has previously been investigated with a related focus on motion segmentation [20]. For example, Kang et al. [21] have recently shown an improvement in the state of the art while using a structure consistency constraint similar to one of our motion cues. Leibe et al. [22] address recognition of cars and pedestrians from a moving vehicle. Our technique handles both these and nine further categories, and additionally semantically segments the image, without requiring their expensive stereo setup.

Optical flow has aided recognition of objects for static cameras [23], but forward ego-motion dominates the visual changes in our footage. Depth-specific motion compensation may help, but requires accurate dense-stereo reconstruction or laser range-scanning. We instead employ features based on a sparse SfM point cloud and avoid these problems.

2 Structure from Motion Point Clouds

We use standard structure from ego-motion techniques to automatically generate a 3D point cloud from video sequences filmed from moving cars. The dominant motion in the sequences gives the camera world-pose and thereby the relative 3D point cloud of all tracked 2D features, including outliers.

We start by tracking 2D image features. Specifically, we use Harris-Stephens corners [24] with localized normalized cross correlation to track 20×20 pixel patches through time in a search window 15% of the image dimensions. In practice, this produced reliable 2D trajectories that usually spanned more than 5 frames. To reduce the number of mis-tracks, each initial template is tracked only until its correlation falls below 0.97.

Footage is obtained from a car-mounted camera. We assume, for purposes of 3D reconstruction, that changes between images are the result of only ego-motion. This allows us to compute a single world-point $W = (x, y, z, 1)^T$ for each point tracked in 2D image space, (u_t, v_t). A best-fit $\tilde{W}$ is computed given at least two corresponding 3x4 camera projection matrices P_t from the sequence. Matrices P are inferred in a robust pre-processing stage, for which we simply use a commercial product [4], which normalizes the resulting up-to-scale solutions to 1.0. Then P is split into row vectors $p_{1:3}$, so W projects into the camera C_t as $[u_1, v_1]^T \equiv [\tilde{u}_1, \tilde{v}_1, \lambda]^T = [p_1, p_2, p_3]^T [x, y, z]^T$, and dividing through by λ gives $u_1 = \frac{p_1 W}{p_3 W}, v_1 = \frac{p_2 W}{p_3 W}$, and similarly for (u_2, v_2), P_{t+1}, and C_{t+1}. As long as the

feature was moving, a least squares solution exists for the three unknowns of $\tilde{W}$, given these four or more (in the case of longer feature tracks) equations. We reconstruct using only the most temporally separated matrices P, instead of finding a $\tilde{W}$ based on the whole 2D track. This strategy generally gives maximum disparity and saves needless computations. After computing the camera poses, no outlier rejection is performed, so that an order of magnitude more tracked points are triangulated for the point cloud.

3 Motion and 3D Structure Features

We now describe the new motion and 3D structure features that are based on the inferred 3D point cloud. We suggest five simple cues that can be estimated reliably and are projected from the 3D world into features on the 2D image plane, where they enable semantic segmentation. We conclude the section by explaining how a randomized decision forest combines these simple weak features into a powerful classifier that performs the segmentation.

3.1 Cues from Point Clouds

Just as there are many ways to parameterize the colors and texture of appearance, there are numerous ways to parameterize 3D structure and motion. We propose five motion and structure cues. These are based on the inferred 3D point cloud, which, given the small baseline changes, is rather noisy. The cues were chosen as robust, intuitive, efficient to compute, and general-purpose but object-category covariant, though these five are by no means exhaustive. The cues also fit nicely with the powerful 3D to 2D projection mechanism (Sect. 3.2). With the driving application in mind, they were designed to be invariant to camera pitch, yaw, and perspective distortion, and could generalize to other domains.

The cues are: height above the camera, distance to the camera path, projected surface orientation, feature track density, and residual reconstruction error. These are intentionally weak; stronger features would not work with the sparse noisy point clouds, though dense feature tracking could someday enable one to apply [25]. We use machine learning to isolate reliable patterns and build a strong classifier that combines many of these cues (Sect. 3.3). By projecting from the 3D point cloud to the 2D image as described in Sect. 3.2, we are able to exploit contextual relationships. One of the benefits of video is that analysis of one frame can often be improved through information in neighboring frames. Our cues take advantage of this since feature tracks exist over several frames.

Height above the camera f_{H}. During video of a typical drive, one will notice that the only fairly fixed relationship between the 3D coordinate frames of the camera C and the world is the camera's height above the pavement (Fig. 2). Measuring height in image-space would be very susceptible to bumps in the road. Instead, after aligning the car's initial "up" vector as the camera's $-y$ axis, the height of each world point $\tilde{W}$ is compared to the camera center's y coordinate as $f_{\mathrm{H}}(\tilde{W}) = \tilde{W}_y - C_y$. By including a fixed offset C_y, the algorithm

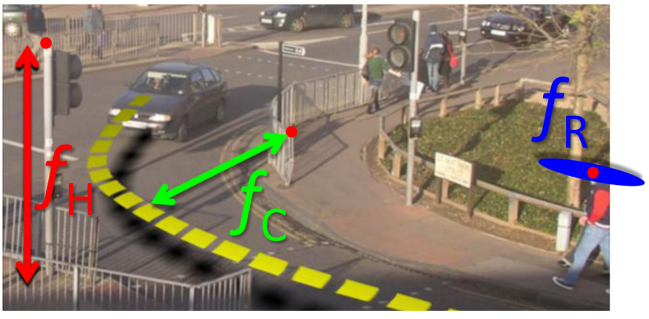


Fig. 2. The height, camera distance, and residual error features are illustrated for a car following the dotted yellow path. The red vertical arrow shows how f_H captures the height above the ground of a 3D point (red dot) reconstructed at the top of the stop light. The green arrow reflects the smallest distance between the point on the railing and the car's path. The blue ellipse for f_R illustrates the large residual error, itself a feature, in estimating the world coordinate $\tilde{W}$ of a point on the moving person's head.

can be trained on point clouds from one vehicle, but run on other cameras and vehicles. Our experiments use footage from two different cars.

Closest distance to camera path f_C. The paths of moving vehicles on road surfaces are less repeatable than a class's absolute height in world coordinates, but classes such as buildings and trees are normally set back from driving roads by a fixed distance (Fig. 2). This feature, using the full sequence of camera centers $C(t)$, gives the value of the smallest recorded 3D separation between C and each $\tilde{W}$ as $f_C(\tilde{W}) = \min_t \|\tilde{W} - C(t)\|$. Note that the smallest separation may occur after a feature in the current frame goes out of view. Such is the case most obviously with features reconstructed on the surface of the road.

Surface Orientation f_{O_x}, f_{O_y}. The points $\tilde{W}$ in the point cloud are too sparse and inaccurate in depth to allow an accurate 3D reconstruction of a faceted world, but do still contain useful spatial information. A 2D Delaunay triangulation [26] is performed on all the projected $\tilde{W}$ points in a given frame. Each 2D triangle is made of 3D coordinates which have inaccurate depths but, heuristically, acceptable relative depth estimates, and thus can give an approximate local surface orientation. The 3D normal vector for each triangle is projected to an angled vector on the image plane in 2D. The x and y components of this 2D angle are encoded in the red and green channels of a false-rendering of the triangulation, shown in the supplementary data online.

Track Density f_D. Faster moving objects, like oncoming traffic and people, often yield sparser feature tracks than stationary objects. Further, some object classes have more texture than others.We thus use the track density as one of the motion-derived cues. $f_D(t)$ is the 2D image-space map of the feature density, *i.e.,* features with the requisite lifespan (3 frames) that were being tracked at a given time. For example, buildings and vegetation have high density, roads and sky have low density, and cars have both types of regions locally.

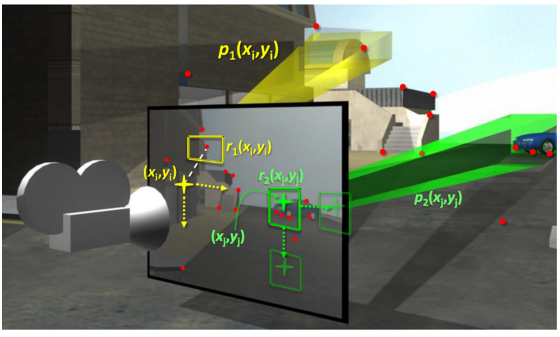


Fig. 3. Points in the 3D point cloud are marked as red dots, as are their projections from world space to the camera's image plane. Any feature information associated with a 3D point also lands on the image plane and is summed in Equations 1, 2 or 3. The yellow and green crosses illustrate how the algorithm slides over each pixel in turn to classify it using a randomized decision forest. Feature responses are calculated at a fixed relative 2D offset (white dashed line) and rectangle r. Here we show two example rectangles r_1 (yellow) and r_2 (green) with their associated truncated pyramids p_1 and p_2. Rectangle r_1 is offset up and to the left of pixel (x_i, y_i), and thus can use the context of *e.g.,* f_{C} to help determine the category at (x_i, y_i). Rectangle r_2 is centered on pixel (x_j, y_j) (*i.e.,* no offset), and thus pools the local information of *e.g.,* $f_{\mathrm{O_x}}$.

Backprojection Residual f_{R}. Having computed a 3D position $\tilde{W}$ for each trajectory (u_t, v_t), we compute $q(\tilde{W})$, the 2D variance of its reprojection error with respect to that track in pixels (Fig. 2). This serves to measure the accuracy of the rigid-world assumption, and highlights objects that move. We use a logarithmic scaling $f_{\mathrm{R}}(\tilde{W}) = log(1+q(\tilde{W}))$ to prevent apparent corners and tracking errors on distant objects from dominating the residuals caused by real moving objects. This motion-covariant feature is naturally dependent on the extent to which objects move, so should help separate buildings from cars, for example.[2] This cue is illustrated in the supplementary video.

3.2 Projecting from 3D to 2D

We extend the features suggested in [27] to project our cues from the 3D point cloud to the 2D image plane, illustrated in Fig. 3. A classifier is trained to compute a segmentation output for each pixel, scanning across the image. When classifying pixel (x, y) in the image, the randomized decision forest, described in Sect. 3.3, computes feature responses using rectangles $r(x, y)$ defined relative to (x, y). Given the camera center, each 2D rectangle implicitly defines a 3D truncated pyramid $p(x, y)$ forward of the image plane. For visible 3D world points within a truncated pyramid, the cue values are summed to give the feature responses, as follows. For heights f_{H}, camera path distances f_{C}, and residuals f_{R} the response is calculated as:

[2] Of course, however, it may also separate parked cars from moving ones.

$$F_T(x,y) = \sum_{\tilde{W} \in p(x,y)} f_T(\tilde{W}) \quad \text{for} \quad T \in \{\mathrm{H}, \mathrm{C}, \mathrm{R}\}. \tag{1}$$

For surface orientation, the triangulated mesh is projected directly into the image, and the sum is over image pixels rather than world points:

$$F_{\mathrm{O_x}}(x,y) = \sum_{(x',y') \in r(x,y)} f_{\mathrm{O_x}}(x',y'), \tag{2}$$

and similarly for $F_{\mathrm{O_y}}$. For track density, the response is

$$F_{\mathrm{D}}(x,y) = |\{\tilde{W} \in p(x,y)\}|, \tag{3}$$

i.e., the number of tracked points within pyramid p. Given this projection, we can make use of integral images [28] in the image plane, one for each cue, for fast feature response computation.

By defining the rectangles (and thereby truncated pyramids) relative to pixel (x, y), we can capture contextual relationships. For example, when classifying for a car pixel, it may be useful to know that a rectangle under the car has a road-like structure (see Fig. 3).

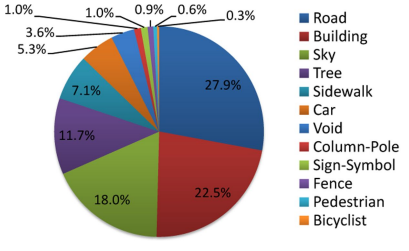
3.3 Randomized Forest

Recent work [7] has employed randomized decision forests for fast and accurate segmentation using appearance features. We implemented a similar randomized forest classifier for segmentation based on our motion and structure features. It serves as a simple to implement and fast algorithm, that crucially, allows us to compare our motion and structure cues to the newest appearance results, on a level playing field. A number of randomized decision trees are averaged together to achieve robust segmentation and avoid over-fitting [29]. Each decision tree recursively branches down from root to leaf nodes. The non-leaf nodes compare a feature response F from (1), (2) or (3) to a learned threshold. At the leaf nodes, there is a class distribution learned from the training data, implicitly sharing features between classes. The MAP classification is given as the segmentation at each pixel. We use the extremely randomized trees algorithm [30] to train the forests. This recursively splits the training data, taking at each split the feature and threshold that maximizes the expected gain in information about the node categories. We follow the idea suggested in [7] of balancing the categories to optimize the category average performance versus the global performance.

4 Experiments

The extensive experiments evaluated whether the simple ego-motion-derived cues could perform object recognition and segmentation. Since no existing database met those needs, we created a new labeled dataset of driving sequences. We then evaluated our motion and structure cues and compare them to existing appearance-based features. We finally show how our motion and structure cues can be combined with these appearance cues to improve overall performance. Further results including videos are available online.

Data Acquisition. Existing databases of labeled images do not include frames taken from video sequences, and usually label relevant classes with only bounding boxes. It takes the same amount of human effort to semantically label the pixels of N images drawn from video sequences as is needed for N independent photographs. The difference is that in the case of video, each labeled frame could have potentially many other temporally related images associated with it. Without an existing corpus of such data, we proceeded to film 55 minutes of daytime footage, 31 minutes of footage at dusk. Pedestrians and cyclists are visible at almost all times, but usually occupy only a small proportion of the field of view (see Fig. 4 left). The footage includes a variety of urban, residential, and mixed use roads. We developed a special purpose labeling tool for use in hand-segmenting the images. This is essentially a paint program with various edge detection and flood filling capabilities, but it also logs the amount of time and order of paint strokes a user employed to label each class. This data will be publicly available and we anticipate this will be of use to the community.



	# Frames	Dataset
Seq 1	101	Day Train
Seq 2	204	Day Train
Seq 3	62	Dusk Train
Seq 4	171	Day Test
Seq 5	62	Dusk Test

Fig. 4. Left: Breakdown by category (listed clockwise from 12 o'clock) of the proportion of pixels in the 600 manually segmented frames in our driving video database. Right: 30Hz high-definition videos for which every 30th frame was painted manually with per-pixel semantic labels. Sequences were used as either training or testing data.

We selected daytime and dusk sequences, as listed in Fig. 4's table. Labeled images for each set are available at 1 fps, and ego-motion features and camera poses were computed at 30 fps. The labeled data has 11 categories: Building, Tree, Sky, Car, Sign-Symbol, Road, Pedestrian, Fence, Column-Pole, Sidewalk, and Bicyclist. There is also a small number of 'void' pixels not belonging to one of these classes that are ignored.

Accuracy is computed by comparing the ground truth pixels to the inferred segmentation. We report per-class accuracies (the normalized diagonal of the pixel-wise confusion matrix), the class average accuracy, and the global segmentation accuracy. The average accuracy measure applies equal importance to all 11 classes, despite the widely varying class prevalences (Fig. 4 left), and is thus a much harder performance metric than the global accuracy measure. As a baseline for comparison with our results below, chance would achieve a global accuracy of about 9%. This rises to about 20% if the baseline chooses randomly according to the category priors.

Table 1. Results in pixel-wise percentage accuracy on all three training and both test sequences, including both day and dusk frames. Note that (i) accurate semantic segmentation is possible using only motion and structure features, without any appearance information, and (ii) by combining our new motion and structure features with existing appearance features, we obtain a small but significant improvement. See text for more analysis.

	Building	Tree	Sky	Car	Sign-Symbol	Road	Pedestrian	Fence	Column-Pole	Sidewalk	Bicyclist	**Average**	**Global**
Mot & Struct	43.9	46.2	79.5	44.6	19.5	82.5	24.4	**58.8**	0.1	**61.8**	18.0	43.6	61.8
Appearance	38.7	60.7	**90.1**	**71.1**	**51.4**	88.6	**54.6**	40.1	**1.1**	55.5	**23.6**	52.3	66.5
Combined	**46.2**	**61.9**	89.7	68.6	42.9	**89.5**	53.6	46.6	0.7	60.5	22.5	**53.0**	**69.1**

4.1 Testing Motion and Structure Features

We trained a randomized decision forest based on our five motion and structure cues, using combined day and dusk sequences for both training and testing. The results are shown in the top row of Table 1 and the middle row of Fig. 7. These show the main contribution of the paper: that using only motion and structure information derived from sparse and noisy point clouds (Fig. 1), one can accurately segment images from video sequences and recognize the categories present. Observe in Figs. 1 and 7 that our algorithm segments the global scene well and even recognizes some of the smaller classes (*e.g.,* bicycle, sign). In terms of global accuracy, 61.8% of pixels are classified, and the strong average accuracy of 43.6% shows good consistency across the different categories. The perhaps low raw numbers highlight the difficulty of our new data set, but as we discuss shortly are comparable to a state-of-the-art appearance algorithm.

One by-product of balancing the categories during training is that the areas of smaller classes in the images tend to be overestimated, spilling out into the background (*e.g.,* the bicycle in Fig. 7). This suggests a shortcoming of the segmentation forest algorithm suggested in [7], that all pixels of a certain class are treated equally. The method in [31] may help with this. There is also considerable confusion between fence and building which we believe to be shortcomings in the ground truth.

To determine the relative importance of the five motion and structure cues, we analyzed the proportion of each chosen by the learning algorithm, as a function of depth in the randomized forest. In Fig. 5 we observe near the tree roots that there is some bias toward the density, height, and closest distance cues. Further down the tree however, all five cues play an important and balanced role (normals were split into x and y components in the figure). This suggests that the density, height, and closest distance cues work well to segment the rough global structure of the scene, and that the finer details are tackled more consistently by all five cues.

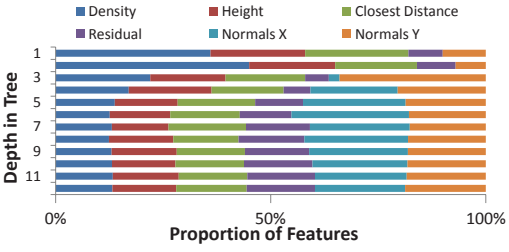


Fig. 5. Proportions of features used in the randomized segmentation forest, as a function of node depth. At the top of the tree there is some bias toward our density, height and closest distance cues. But deeper in the tree all cues are informative and used in roughly equal proportions.

Cues Used	Balanced Ave. Score	Global Score
All	**43.3%**	**63.0%**
Just Height	39.1%	55.3%
Just Distance	41.9%	57.1%
Just Orient.	37.3%	59.0%
Just Density	40.2%	60.0%
Just Residual	36.2%	58.1%

Fig. 6. We combine all the cues, but here each cue was also tested in isolation. Scores were computed by either optimizing the balanced per-category average, or the global % correct - of pixels assigned to the same class as in the ground truth.

These result used a randomized forest containing 50 trees trained to a maximum depth of 13, testing 500 random features (cue choice and offset rectangles) at each step of building the tree. The learning takes only about 15 minutes, and testing takes less than one second per frame.[3] Our system should scale well, at worst linearly with the number of object classes and training images.

4.2 Comparison with Appearance Features

We compared with a state-of-the-art technique [7]. It uses dense pixel patches to semantically segment images using only appearance information (no motion or structure). Table 1 includes the comparison between our motion and structure features vs. the appearance features of [7]. As one might expect, given much denser and less noisy image features, appearance works somewhat better than motion and structure, though clearly this does not diminish our contribution that the new motion and structure cues work at all. We discuss below how these two complementary types of feature can be combined to improve overall results.

Motion and structure features do however have an obvious advantage over appearance features: generalization to novel lighting and weather conditions. We compare in Table 2 the global and average segmentation accuracies obtained when training in one lighting condition (day or dusk) and testing in the other. Figure 8 and the online materials show segmentation results. We see for both combinations that the new motion and structure features generalize much better than the appearance features. Extra labeled data could be used to improve the appearance features, but obtaining labeled data is very expensive. Without *any* extra data, our motion and structure features can reasonably be expected to

[3] These timings assume pre-computed SfM point clouds. Recent work [22] has moved towards making this real-time.

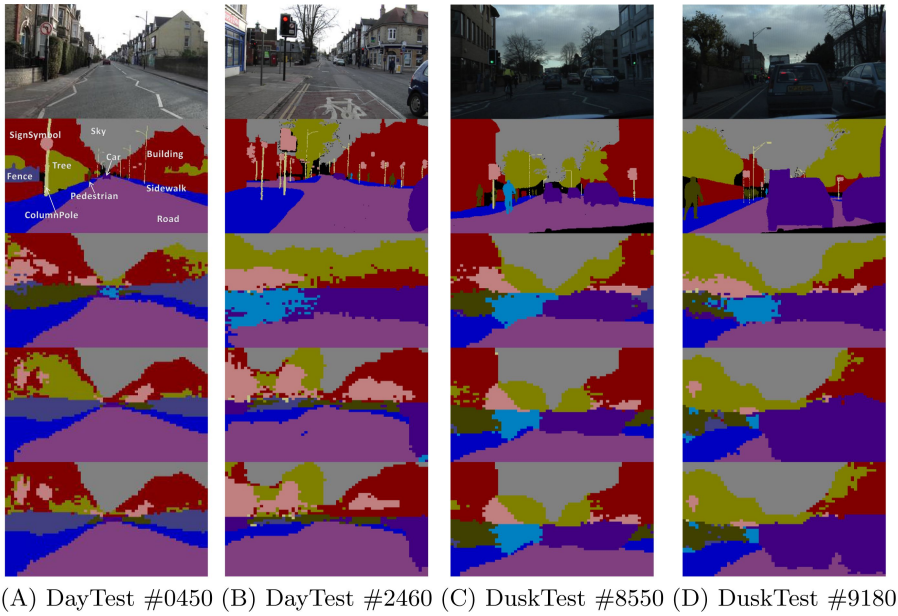
Table 2. By training in one lighting condition (day or dusk) and testing in the other, we compare the lighting invariance of our motion and structure features with appearance based features. Observe much better generalization of our motion and structure features to novel lighting conditions.

	Train Day – Test Dusk		Train Dusk – Test Day	
	Average	Global	Average	Global
Mot & Struct	**29.2%**	**45.5%**	**31.0%**	**59.4%**
Appearance	14.2%	21.7%	25.4%	50.5%

generalize to other lighting and weather conditions such as night, snow or rain, since they are almost independent of image appearance (up to obtaining feature tracks).

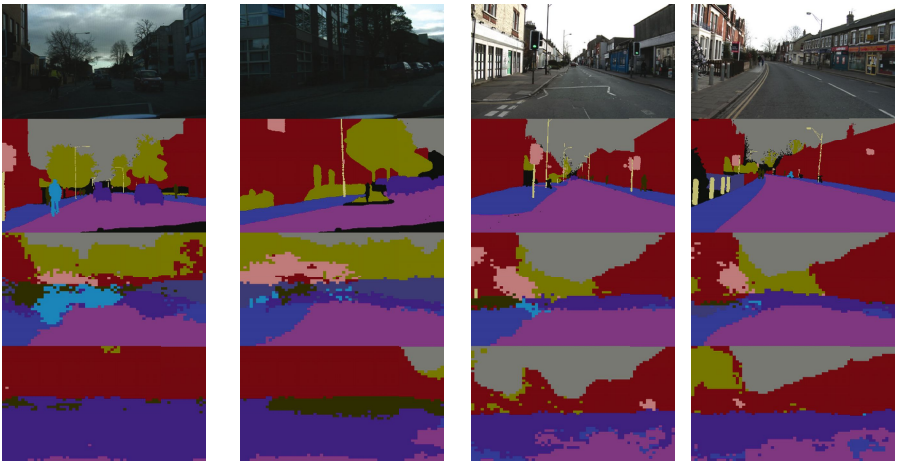
4.3 Combined Ego-Motion and Texton Features

Since our motion and structure features contain rather different information to the appearance features of [7], one would expect the two to be complementary. We investigated a simple method of combining the features, by taking a geometric interpolation of the two classifiers. We denote our randomized decision forest



(A) DayTest #0450 (B) DayTest #2460 (C) DuskTest #8550 (D) DuskTest #9180

Fig. 7. Sample segmentation results. From top to bottom: test image, ground truth, motion and structure inferred segmentation, appearance inferred segmentation, and combined segmentation. Note that accurate segmentation and recognition is possible using only motion and structure features, and that combining our new cues with existing appearance cues gives improved segmentation. The whole video sequence is online.



(A) DuskTest #8580 (B) DuskTest #10020 (C) DayTest #0960 (D) DayTest #4680

Fig. 8. Inherent invariance of motion and structure to novel lighting conditions. From top to bottom: test image, ground truth, motion and structure inferred segmentation, and appearance inferred segmentation. When trained on daytime footage and tested on dusk footage and vice-versa, our motion and structure cues are still able to accurately recognize and segment the scene. In contrast, the appearance inferred segmentation degrades drastically.

classifier based on motion and structure cues as $P(c|M)$, and the appearance-based classifier from [7] as $P(c|A)$. These were trained independently and then combined as

$$P(c_{(x,y)}|M,A) = \frac{1}{Z}\ P(c_{(x,y)}|M) \times P(c_{(x,y)}|A)^{\alpha}, \tag{4}$$

where α is a weighting parameter chosen by holdout validation, and Z is used to renormalize the distribution. The two distributions $P(c|M)$ and $P(c|A)$ should reinforce their decisions when they agree and flatten the distribution when they disagree, a kind of soft 'AND' operation. This was found better in practice than an arithmetic average ('OR').

The results for this combination can be seen in the last row of Table 1 and Fig. 7, and in the online video, using $\alpha = 2.5$. The soft AND operation does not guarantee an improvement for all categories, but still we observe a small but significant improvement in both average and global accuracy. The qualitative appearance of the segmentations is also consistently improved. These results are very encouraging, suggesting that our motion and structure features are indeed complementary to appearance features.

5 Conclusions

Using motion and 3D world structure for segmentation and object recognition is a fundamentally new challenge. Our main contribution has been to show that

accurate results are possible using only ego-motion derived 3D points clouds. Experiments on a challenging new database of naturally complex driving scenes demonstrate that our five new motion and structure cues can be combined in a randomized decision forest to perform accurate semantic segmentation. These five cues were also shown to generalize better to novel lighting conditions than existing appearance-based features. By then combining motion and structure with appearance, an overall quantitative and qualitative improvement was observed, above what either could achieve individually.

The worst performance of our system is for those categories least well represented in the training data, despite balancing categories during training. We hope that semi-supervised techniques that use extra partially labeled or unlabeled training data may lead to improved performance in the future.

Our combination of segmentation classifiers (Equation 4) is somewhat simplistic, and we are investigating other methods. Learning a histogram for each pair of (motion and structure, appearance) tree leaf nodes could better model the joint dependencies of the two classifiers, but care must be taken so that in avoiding overfitting, quadratically more training data is not required.

Acknowledgements. Thanks to John Winn for advice and for driving one of the capture cars.

References

1. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
2. Merrell, P., Akbarzadeh, A., Wang, L., Mordohai, P., Frahm, J.M., Yang, R., Nister, D., Pollefeys, M.: Real-time visibility-based fusion of depth maps. In: Proceedings of the International Conference on Computer Vision (ICCV) (2007)
3. Posner, I., Schroeter, D., Newman, P.M.: Describing composite urban workspaces. In: ICRA (2007)
4. Boujou: 2d3 Ltd. (2007), `http://www.2d3.com`
5. Chum, O., Zisserman, A.: An exemplar model for learning object classes. In: CVPR (2007)
6. Li, L.J., Fei-Fei, L.: What, where and who? classifying events by scene and object recognition. In: ICCV (2007)
7. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: CVPR (2008)
8. Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. In: CVPR, vol. 2, pp. 2137–2144 (2006)
9. Hoiem, D., Efros, A.A., Hebert, M.: Geometric context from a single image. In: ICCV, vol. 1, pp. 654–661 (2005)
10. Huber, D., Kapuria, A., Donamukkala, R., Hebert, M.: Parts-based 3d object classification. In: CVPR, pp. 82–89 (2004)
11. Hoiem, D., Rother, C., Winn, J.: 3d layout crf for multi-view object class recognition and segmentation. In: CVPR (2007)
12. Kushal, A., Schmid, C., Ponce, J.: Flexible object models for category-level 3d object recognition. In: CVPR (2007)

13. Pingkun, Y., Khan, S., Shah, M.: 3d model based object class detection in an arbitrary view. In: ICCV (2007)
14. Savarese, S., Fei-Fei, L.: 3d generic object categorization, localization and pose estimation. In: ICCV (2007)
15. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 428–441. Springer, Heidelberg (2006)
16. Cedras, C., Shah, M.: Motion-based recognition: A survey. IVC 13(2), 129–155 (1995)
17. Laptev, I., Lindeberg, T.: Space-time interest points. In: ICCV, pp. 432–439 (2003)
18. Viola, P.A., Jones, M.J., Snow, D.: Detecting pedestrians using patterns of motion and appearance. In: ICCV, pp. 734–741 (2003)
19. Yin, P., Criminisi, A., Winn, J.M., Essa, I.: Tree-based classifiers for bilayer video segmentation. In: CVPR (2007)
20. Wiles, C., Brady, M.: Closing the loop on multiple motions. In: ICCV, pp. 308–313 (1995)
21. Kang, J., Cohen, I., Medioni, G.G., Yuan, C.: Detection and tracking of moving objects from a moving platform in presence of strong parallax. In: ICCV, pp. 10–17 (2005)
22. Leibe, B., Cornelis, N., Cornelis, K., Gool, L.J.V.: Dynamic 3d scene analysis from a moving vehicle. In: CVPR (2007)
23. Efros, A.A., Berg, A.C., Mori, G., Malik, J.: Recognizing action at a distance. In: ICCV, pp. 726–733 (2003)
24. Harris, C., Stephens, M.: A Combined Corner and Edge Detector. In: 4th ALVEY Vision Conference, pp. 147–151 (1988)
25. Mitra, N.J., Nguyen, A., Guibas, L.: Estimating surface normals in noisy point cloud data. International Journal of Computational Geometry and Applications 14, 261–276 (2004)
26. Shewchuk, J.R.: Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In: Lin, M.C., Manocha, D. (eds.) FCRC-WS 1996 and WACG 1996. LNCS, vol. 1148, pp. 203–222. Springer, Heidelberg (1996)
27. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 1–15. Springer, Heidelberg (2006)
28. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR, vol. 1, pp. 511–518 (2001)
29. Amit, Y., Geman, D.: Shape quantization and recognition with randomized trees. Neural Computation 9(7), 1545–1588 (1997)
30. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine Learning 36(1), 3–42 (2006)
31. Winn, J., Shotton, J.: The layout consistent random field for recognizing and segmenting partially occluded objects. In: CVPR (2006)

Keypoint Signatures for Fast Learning and Recognition*

Michael Calonder, Vincent Lepetit, and Pascal Fua

Computer Vision Laboratory, EPFL, Switzerland
{michael.calonder,vincent.lepetit,pascal.fua}@epfl.ch

Abstract. Statistical learning techniques have been used to dramatically speed-up keypoint matching by training a classifier to recognize a specific set of keypoints. However, the training itself is usually relatively slow and performed offline. Although methods have recently been proposed to train the classifier online, they can only learn a very limited number of new keypoints. This represents a handicap for real-time applications, such as Simultaneous Localization and Mapping (SLAM), which require incremental addition of arbitrary numbers of keypoints as they become visible.

In this paper, we overcome this limitation and propose a descriptor that can be learned online fast enough to handle virtually unlimited numbers of keypoints. It relies on the fact that if we train a Randomized Tree classifier to recognize a number of keypoints extracted from an image database, all other keypoints can be characterized in terms of their response to these classification trees. This signature is fast to compute and has a discriminative power that is comparable to that of the much slower SIFT descriptor.

1 Introduction

Feature point recognition and matching are crucial for many vision problems, such as pose estimation or object detection. Most current approaches rely on local descriptors designed to be invariant, or at least robust, to affine deformations. Among these, the SIFT descriptor [1] has been shown to be one of the most effective [2] and its recognition performance can be further enhanced by affine rectifying the image patches surrounding the feature points [3]. However, such descriptors are relatively slow and cannot be used to handle large numbers of feature points in real-time.

Faster SIFT-like descriptors such as SURF [4] achieve 3 to 7-fold speed-ups by exploiting the properties of integral images. However, it has recently been shown that even shorter run-times can be obtained without loss in discriminative power by reformulating the matching problem as a classification problem. This approach relies on an offline training phase during which multiple views of the feature points to be matched are used to train a classifier to recognize them based on a few pairwise intensity comparisons [5,6]. The resulting run-time computational complexity is much lower than that of SIFT-like descriptors while

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

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 58–71, 2008.

preserving robustness to viewpoint and lighting changes. However, this comes at the cost of an offline training phase that is relatively slow and ill-adapted to real-time applications, such as Simultaneous Localization and Mapping (SLAM), which require learning the appearance of new feature points as they become visible. As a result, real-time algorithms that rely on this approach keep a very tight bound on the number of keypoints they work with [7].

In this paper, we remove this limitation and show that online learning of keypoint appearance can be made fast enough for a real-time system to handle a virtually unlimited number of keypoints. At the heart of our approach that we will refer to as a Generic Tree (GT) algorithm is the following observation: If we train a Randomized Tree classifier [5] to recognize a number of keypoints extracted from an image database, all other keypoints can be characterized in terms of their response to these classification trees, which we will refer to as their *signature*. Because the signature can be computed very fast, the learning becomes quasi-instantaneous and therefore practical for online applications. We attribute this desirable behavior to the fact that, assuming the initial set of keypoints is rich enough, the new keypoints will be similar to some of those initial points and the signature will summarize these similarities. In other words, we replace the hand-crafted SIFT descriptor by one that has been empirically learned from training data to be very selective. Remarkably, this can be done using a fairly limited number—300 in our experiments—of initial keypoints.

In the remainder of the paper, we first discuss related work. We then describe our method and compare it against state-of-the-art ones on standard benchmark images. Finally, we show that it can be integrated into a SLAM algorithm to achieve real-time performance.

2 Related Work

As discussed in the introduction, state-of-the-art approaches to feature point matching can be classified into two main classes.

Those in the first class rely on local descriptors designed to be invariant, or at least robust, to specific classes of deformations [8,1]. They often require scale and rotation estimates provided by a keypoint detector. Among these, the SIFT descriptor [1], computed from local histograms of gradients, has been shown to work remarkably well, especially if one rectifies the image patches surrounding the feature points [2,3]. We will therefore use it as a benchmark against which we will compare the performance of our approach. However, it must be noted that, because the SIFT descriptor is complex, it is also relatively slow to evaluate. On a modern PC, it takes approximately 1ms per feature point[1], which limits the number of feature points that can be handled simultaneously to less than 50 if

[1] Some commercial implementations of SIFT, such as the one by Evolution Robotics, can be up to 10 times faster by using careful coding and processor extensions, but those are not easily available. Furthermore, this would not represent a fair comparison to our own implementation that does not include any such coding but could be similarly sped-up.

one requires frame-rate performance. SURF [4] is closely related to SIFT and achieves a 3 to 7-fold speed increase by efficiently using integral images and box filters to compute the descriptor, which means that from 150 to 350 keypoints could be handled. By contrast, our approach can compute several thousand signatures at frame-rate.

Of course, SIFT and SURF are nevertheless effective for well-designed real-time applications. For example, it has been shown that feature points can be used as visual words [9] for fast image retrieval in very large image databases [10]. The feature points are labeled by hierarchical k-means clustering of their SIFT descriptors, which allows to use very many visual words. However, the performance is measured in terms of the number of correctly retrieved documents rather than the number of correctly classified feature points. For applications such as pose estimation or SLAM, the latter criterion is much more important.

A second class of approaches to feature point matching relies on statistical learning techniques to compute a probabilistic model of the patches surrounding them. The one-shot approach of [11] uses PCA and Gaussian Mixture Models but does not account for perspective distortion. Since the set of possible appearances of patches around an image feature, seen under changing perspective and lighting conditions, can be treated as a class, it was later shown that a classifier based on Randomized Trees [12] can be trained to recognize them [5] independently of pose. This is done using a database of patches that is obtained by warping keypoints of a reference image by randomly chosen homographies. The resulting algorithm has very fast run-time performance but requires a computationally training phase that precludes online learning of new feature points. This limitation has been partially lifted by optimizing the design of the classifier and exploiting the power of modern graphic cards [7], but still only allows for learning small numbers of new feature points. By contrast, the method we propose here can learn virtually unlimited numbers of new signatures at a very small computational cost.

3 Generic Trees

The idea behind our Generic Trees (GTs) method is to take advantage of a fast classifier to efficiently compute short description vectors, or signatures, for arbitrary keypoints.

Fig. 1. 3 out of 7 landscape images from which the base set of keypoints were extracted

We start from a relatively small set of keypoints that we extracted from images such as those of Fig. 1. We refer to this set as *base set* and train a Randomized Tree classifier to recognize the keypoints in the base set under arbitrary perspective, scale, and lighting conditions [5]. Given a new keypoint that is *not* in the base set, we show below that the classifier responds to it in a way that is also stable to changes in scale, perspective, and lighting. We therefore take this response to be the compact and fast-to-compute signature we are looking for.

3.1 Stable Signatures

More formally, every keypoint $\mathbf{u}_i \in \mathbb{R}^2$ in the beforementioned base set is related to exactly one point $\mathbf{k}_i$ in 3D. Let us start with a set of N points $\mathbf{K} = \{\mathbf{k}_1, \ldots, \mathbf{k}_N\}$, $\mathbf{k}_i \in \mathbb{R}^3$, and refer to N as *base size*. We then build a classifier based on Randomized Trees that is able to recognize the $\mathbf{k}_i$ under varying viewing conditions [5]. Let $\mathbf{p}_i$ be the patch centered on $\mathbf{u}_i$. Then the classifier provides a function $\mathcal{C}(\mathbf{p}_i)$ mapping a patch $\mathbf{p}_i$ to a vector in $\mathbb{R}^N$. Using the notation $\mathcal{C}^{(j)}(\mathbf{p}_i)$ to refer to the j-th element of the vector $\mathcal{C}(\mathbf{p}_i)$, $1 \leq j \leq N$, we can state a special property of $\mathcal{C}$:

$$\mathcal{C}^{(j)}(\mathbf{p}_i) \quad \text{is} \quad \begin{cases} \text{large} & \text{if } j = i \\ \text{small} & \text{otherwise} \end{cases}.$$

This is shown in Fig. 2-LEFT for $i = 250$ and $N = 300$.

Furthermore, let $\mathcal{T}(\mathbf{p}, \boldsymbol{\Theta})$ be a transformation of an image patch $\mathbf{p}$ under viewing condition change $\boldsymbol{\Theta}$. $\boldsymbol{\Theta}$ typically encodes changes in illumination, viewpoint, or scale. If the classifier has been trained well, we can assume that

$$\forall \boldsymbol{\Theta} \quad : \quad \mathcal{C}(\mathbf{p}) \approx \mathcal{C}\big(\mathcal{T}(\mathbf{p}, \boldsymbol{\Theta})\big) \,. \tag{1}$$

When we consider a new 3D-point κ that does *not* belong to $\mathbf{K}$ and center a patch $\mathbf{q}$ on the keypoint corresponding to κ, we can define the signature of the patch $\mathbf{q}$ simply as

$$\mathsf{signature}(\mathbf{q}) = \mathcal{C}(\mathbf{q}).$$

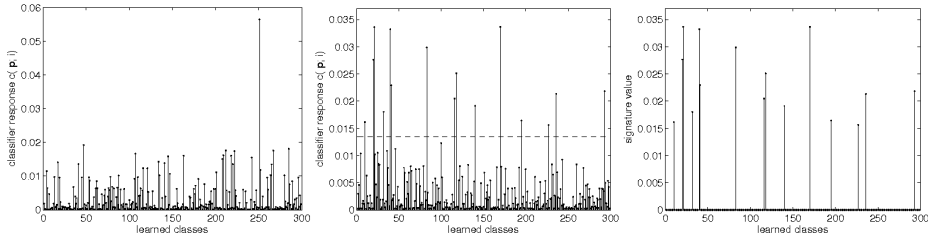


Fig. 2. LEFT: Response of trained Randomized Trees to a patch around a keypoint in the base set, $N = 300$. The response has typically only one spike. MIDDLE: Response to a patch of a keypoint that is *not* in the base set. The response has several peaks which are stable under viewpoint variation. RIGHT: A typical signature. The thresholding process of Eq. 2 takes the raw classifier response and sets all values below t to zero.

A patch $\mathbf{q}'$ centered on the keypoint of κ in another image can be written as $\mathcal{T}(\mathbf{q}, \mathbf{\Theta})$, for some $\mathbf{\Theta}$. Under the assumption of Eq. 1, the signature of $\mathbf{q}'$ is equal to the signature of $\mathbf{q}$ because

$$\mathsf{signature}(\mathbf{q}') = \mathcal{C}(\mathbf{q}') = \mathcal{C}\big(\mathcal{T}(\mathbf{q}, \mathbf{\Theta})\big) = \mathcal{C}(\mathbf{q}) = \mathsf{signature}(\mathbf{q}).$$

In other words, the signature is stable under changes in viewing conditions.

3.2 Sparse Signatures

Because κ is not a member of $\mathbf{K}$, the response of the Trees to the corresponding patch $\mathbf{q}$, $\mathcal{C}(\mathbf{q})$, has typically more than one peak. Such a response is shown in Fig. 2-Middle. In practice, only few of the values in $\mathcal{C}(\mathbf{q})$ are large. We therefore replace the signature above by one which is much sparser:

$$\mathsf{signature}(\mathbf{q}) = \mathrm{th}(\mathcal{C}(\mathbf{q})) = \left[\mathrm{th}(\mathcal{C}^{(1)}(\mathbf{q})), \ldots, \mathrm{th}(\mathcal{C}^{(N)}(\mathbf{q}))\right]^\top, \tag{2}$$

where $\mathrm{th}(\cdot)$ is a thresholding function:

$$\mathrm{th}(x) = \begin{cases} x \text{ if } x \geq t \\ 0 \text{ if } x < t \end{cases}. \tag{3}$$

As shown in Fig. 2-RIGHT, most of the values in the signatures are null and matching reduces to computing the Euclidean distance between two (sparse) signatures. To this end, efficient approaches exist and we use the best-bin-first (BBF) algorithm [13] to match signatures.

The threshold t is a free parameter the user has to provide and it depends on the base size N, because the response of the classifier is normalized. Its impact on the accuracy and the speed of the classifier will become clear in Section 4.

3.3 Base Set and Trees

As discussed above, we extract the base set from landscape images such as those of Fig. 1. It is worth noting that we have tried extracting them from other kinds of images, such as pictures of indoor scenes or animals, which has not resulted in any appreciable change in the performance of our algorithm. In other words, our experiments show that we can extract the base set from almost any kind of image as long as it exhibits enough structure and variety.

In practice, we use the DOG/SIFT feature detector, which typically finds several thousands of keypoints in each image. To create a base set of size N, we *randomly* select N of these keypoints, the only constraint being that they should be at least 5 pixels away from each other.

We then train a Randomized Trees classifier to recognize the resulting keypoints. Fig. 3 illustrates the response of our GTs to arbitrary patches, given this classifier. Roughly speaking, computing their signature amounts to finding the subset of the base set they most resemble to.

Fig. 3. The leftmost image of each row represents a patch from a test image. The remaining images in the same row represent the patches surrounding the 10 keypoints the test patch looks most similar to according to our Randomized Tree classifier, in decreasing similarity order.

Fig. 4. Images for the Wall (LEFT) and Light (RIGHT) datasets

4 Results

We first use three publicly available datasets to characterize the behavior of our GTs and then to compare their performance to SIFT, which is widely acknowledged as one of the most effective descriptor in terms of recognition rate. We then demonstrate that the GTs can be effectively integrated into a SLAM algorithm. The resulting system has the ability to learn the signatures of a virtually unlimited number of keypoints and therefore to detect them at every time step without having to depend on knowing the previous camera pose, thus giving it great robustness to occlusions and abrupt camera motions.

4.1 Performance Evaluation

The three image datasets we used for our experiments –Wall[2], Light[2], and Fountain[3]—are depicted by Figs. 4 and 5. The Wall and Light scenes are planar and the relationship between two images in the database can be expressed by a homography. By constrast the Fountain scene is fully three-dimensional and we have access to an accurate laser-scan that can be used to establish explicit one-to-one correspondences at arbitrary locations.

[2] Available at `http://www.robots.ox.ac.uk/~vgg/research/affine/`
[3] Available at `http://cvlab.epfl.ch/~strecha/multiview/`

Fig. 5. Two 1440 × 960 images from the Fountain dataset

Given several images from the same database, we take $m \leftrightarrow n$ to indicate that we use image m as a reference image from which we extract keypoints that we try to match in image n. For the Wall dataset, we tested $1 \leftrightarrow 2$ and $1 \leftrightarrow 3$, corresponding to a change in camera orientation of 20 and 40 degrees, respectively. From the Light dataset, we tested $1 \leftrightarrow 2$, $1 \leftrightarrow 3$, and $1 \leftrightarrow 4$. From the Fountain dataset, we used $1 \leftrightarrow 2$ and $1 \leftrightarrow 3$.

We define the *recognition rate*, the main performance criterion, as the ratio of the number of correct matches to the total number of interest-points in the reference image. For all tests described in this section, we proceed as follows. We extract a number of DOG/SIFT keypoints from the reference image and compute the coordinates of their corresponding points in the test image using the known geometric relationship between the two. We then compute the SIFT descriptors and GT signatures for both sets of test points and store them in two separate test image databases. Matching a point in the reference image then simply amounts to finding the most similar point in terms of either its descriptor or its signature in the test image database. Note that not detecting interest-points in the test image but using geometry instead prevents repeatability problems of the keypoint detector from influencing our results. Furthermore, since we apply the same procedure for SIFT and for GTs, we do not favor either technique over the other.

Signature Length and Base Size. There are only two parameters in GTs: The threshold t and the base size N. t implicitely determines the *signature length*, that is, the number of non-zero entries in the signature. This length, in turn, has a direct impact on the recognition rate. However, it does *not* impact the time it takes to compute the descriptor, it only affects the time it takes to match: The sparser the signature, the less computation is required to compare a given number of signatures. Hence, t controls the trade-off between the recognition rate and the computational effort one is willing to make. By contrast, increasing N slows down both signature computation and matching.

Fig. 6-LEFT shows the distribution of the signature length computed using 1000 keypoints on the Fountain dataset $1 \leftrightarrow 2$ for $t = 0.01$. This yields a mean signature length of 21.2 with standard deviation 3.9. 95% of the signatures have

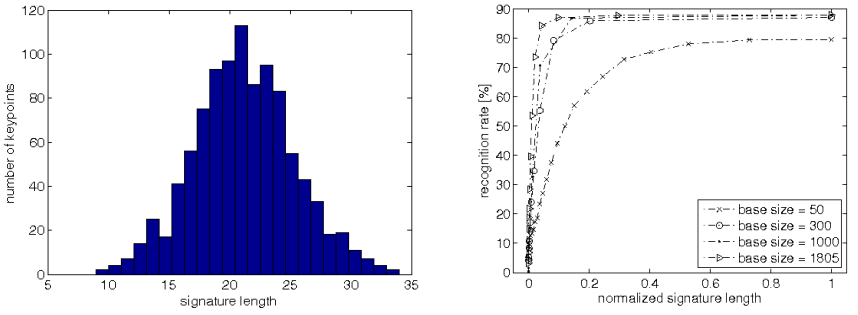


Fig. 6. LEFT: Typical distribution of the GT signature length. The signatures were computed for 1000 keypoints from the Fountain image pair $1 \leftrightarrow 2$. The mean length is 21.2 with a standard deviation of 3.9. RIGHT: Recognition rate as a function of normalized signature length for base sizes ranging from 50 to 1805. The *normalized* signature length is taken to be the ratio of the number of non-zero entries in the signature and the base size N.

a length in the range $\{13, ..., 29\}$ which is a very compact representation. The recognition rate for this case amounts to 75.4%.

In Fig. 6-RIGHT, we plot the recognition rate as a function of the signature length for different values of N, using the Fountain dataset $1 \leftrightarrow 2$. We see that $N = 50$ is too small to achieve the best possible performance of 87.9%, but that going beyond $N = 300$ does not bring any significant improvement. In the remainder of the paper, we will therefore use $N = 300$ for our experiments.

Comparing Recognition Rates. Fig. 7 illustrates our results on the Wall database. On the left, we plot the relationship between the signature length

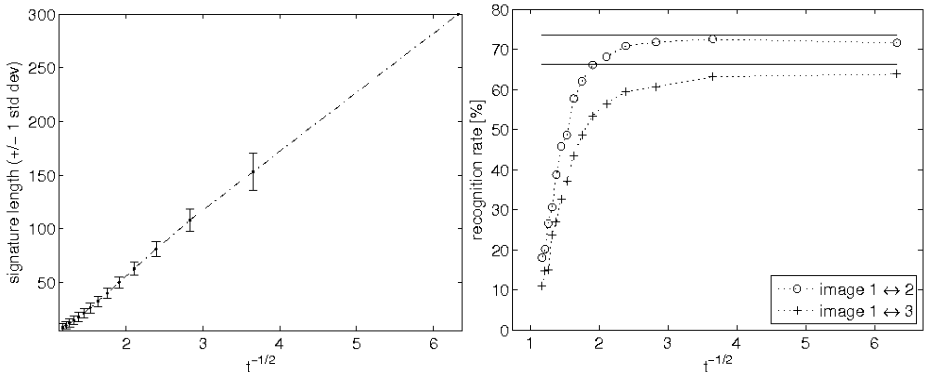


Fig. 7. Wall dataset results. LEFT: Signature length as a function of $\frac{1}{\sqrt{t}}$. The relationship is roughly linear. RIGHT: Recognition rates as a function of $\frac{1}{\sqrt{t}}$. The top curve corresponds to image pair $1 \leftrightarrow 2$, the lower one to $1 \leftrightarrow 3$. The horizontal lines denote the corresponding SIFT results.

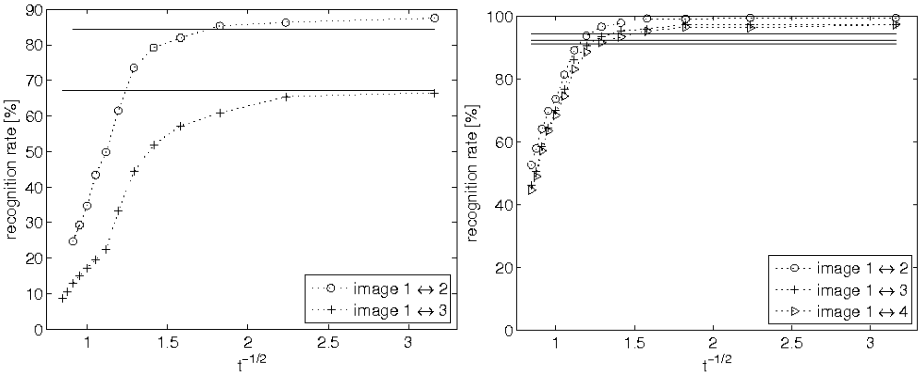


Fig. 8. LEFT: Recognition rates as a function of $\frac{1}{\sqrt{t}}$ for the Fountain dataset. The top curve corresponds to image pair $1 \leftrightarrow 2$ and the lower one to $1 \leftrightarrow 3$. RIGHT: Recognition rates as a function of $\frac{1}{\sqrt{t}}$ for the Light dataset. The three curves correspond to image pairs $1 \leftrightarrow 2$, $1 \leftrightarrow 3$, and $1 \leftrightarrow 4$. The horizontal lines denote the corresponding SIFT results.

and $\frac{1}{\sqrt{t}}$, which is roughly linear. On the right, we plot the performance of GTs as a function of $\frac{1}{\sqrt{t}}$ for two image pairs and we represent the corresponding performance of SIFT by a horizontal line. In Fig. 8, we plot similar graphs for the Fountain and Light datasets. Note that in all cases, GTs rapidly reach their peak performance as $\frac{1}{\sqrt{t}}$ increases, that is, as t decreases. In practice, this means that as long as t is not taken to be too large, its exact value has little influence on the algorithm's recognition performance.

GTs perform a bit better than SIFT on the Light dataset, slightly worse on the Wall dataset, and almost identically on the Fountain dataset. In other words, in terms of recognition performance, GTs and SIFT are almost equivalent.

Comparing Computation Times. Performing a completely fair comparison between the SIFT and GTs is non-trivial. SIFT re-uses intermediate data from the keypoint extraction to compute canonic scale and orientations and the descriptors, while Randomized Trees only require keypoint locations and can therefore work with arbitrary detectors. On the other hand, the distributed SIFT C code is not optimized for the very last cycle[4]. However, the level of optimization of our GT code being roughly comparable, we regard this as a fair comparison. In [6], the authors detail what makes the original Trees much faster at classification in comparison to SIFT. The arguments they put forth carry directly over to the GTs.

Here we compare the CPU time SIFT and GTs require to compute the descriptors.[5] Postprocessing steps are excluded, in particular matching, it can be

[4] In the author's words: "[SIFT was] implemented as efficiently as possible while still maintaining intuitive code".

[5] All experiments were run on a Single-Threaded 2 GHz Intel Xeon machine.

Task	Time
Gaussian pyramid	1434 ms
DOG pyramid	277.4 ms
Feature scales	0.2362 ms
Feature orientations	91.34 ms
Assemble final descriptor	339.2 ms
Total time	2142 ms

(a) SIFT

Task	Time
Keypoint detection (est'd)	5 ms
Compute base distribution	33.21 ms
Thresholding	1.217 ms
Total time	39.43 ms

(b) GTs

Fig. 9. Total time and time required by substeps in SIFT and GTs. The entry *Keypoint detection* in (b) has been added to make the comparison more fair. The value was estimated from [14].

done efficiently for both SIFT descriptors and GT signatures as discussed in Section 3. The times are given in Fig. 9 for 1000 SIFT keypoints on the Fountain dataset. Overall, we obtain a 35-fold speedup for the GTs with respect to SIFT. Note that it would *not* make much sense to compare only the time for feature vector computation of SIFT against the total time of GTs since SIFT cannot forgo the preprocessing stage that computes the orientation and scale of the features. However, to ensure fairness, we added a "virtual" amount of time on the GT side to account for the time spent by an efficient detection algorithm such as FAST [14] to detect 1000 interest-points.

As discussed earlier, faster SIFT-like detectors such as SURF [4] have been proposed and yield a 3 to 7-fold speed increase, which still leaves GTs with a 5 to 11-fold speed advantage.

4.2 SLAM Using Generic Trees

In this section we demonstrate that GTs can increase the robustness of a Visual SLAM system. Their role is twofold. We first use them to bootstrap the system by localizing the camera with respect to a known planar pattern in the scene. Second, we incrementally train a second set of GTs to recognize new landmarks reconstructed by the system. They make the system robust against severe disturbances such as complete occlusion or strong shaking of the camera, as evidenced by the smoothness of the recovered camera trajectory in Fig. 10.

For simplicity, we use a FastSLAM [15,16] approach with a single particle to model the distribution over the camera trajectory. This is therefore a simplified version of FastSLAM, but our GTs are sufficiently powerful to make it robust.

As discussed above, we use two different sets of GTs. We will refer to the first set as "Offline GTs", that we trained offline to recognize keypoints on the planar pattern. It is visible in the first frame of the sequence to bootstrap the system and replaces the four fiducials used by many other systems. We show the initialization process in Fig. 10 (a). This increases the flexibility of our system since we can use any pattern we want provided that it is textured enough. The

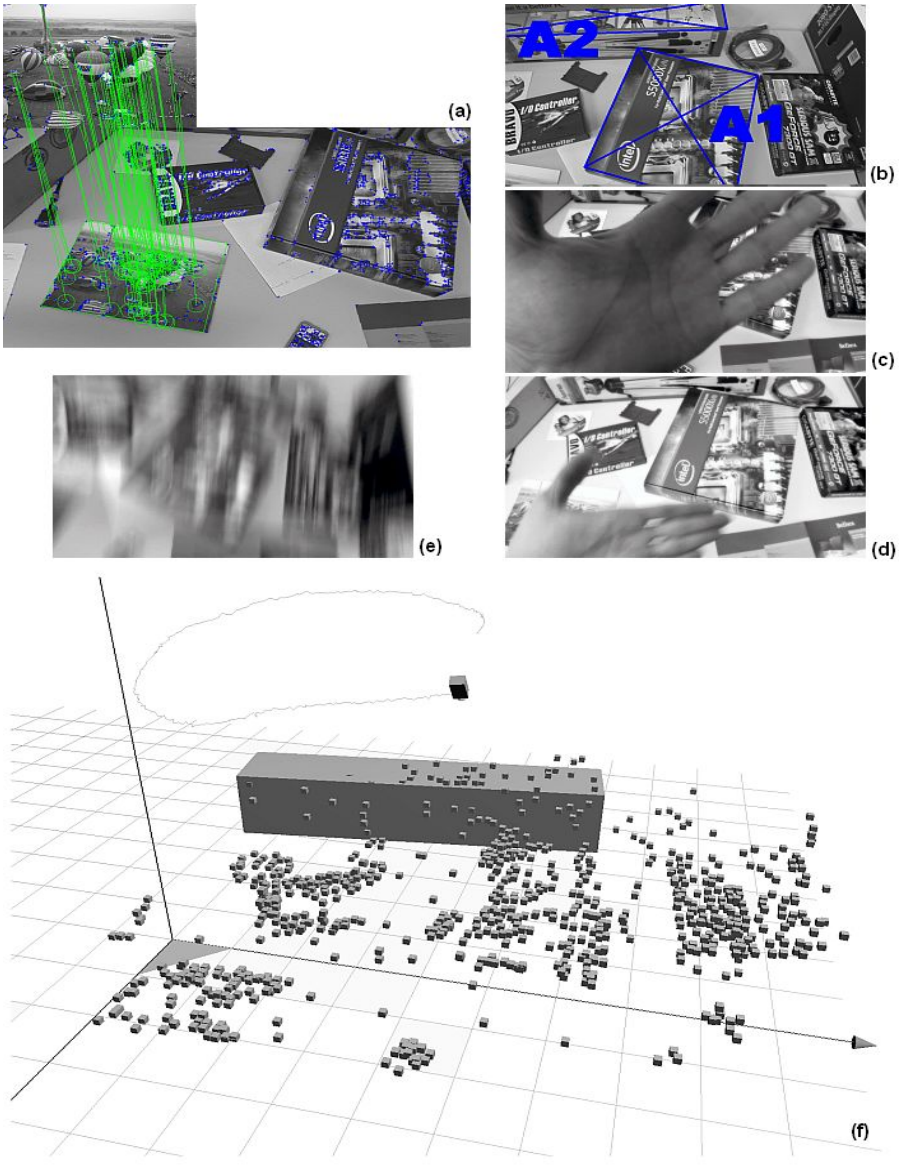


Fig. 10. GTs applied to SLAM. (a) Initialization: The pattern at the top is detected in the image yielding initial landmarks and pose. (b-e) Four images from the sequence, which features both partial/total occlusions, strong camera shaking, and motion blur. (f) The reconstructed trajectory and landmarks. The latter are depicted as cubes centered around the current state of the corresponding EKF. Note how smooth the trajectory is, given the difficulties the algorithm had to face. The rectangular box corresponds to a true object in the scene and was inserted manually in order to support the visual inspection of the scene.

second set of GTs, the "Online GTs", are incrementally trained[6] to recognize the 3D landmarks the SLAM system discovers and reconstructs.

Our complete algorithm goes through the following steps:

1) Initialize the camera pose and some landmarks by detecting a known pattern using the Offline GTs.
2) Detect keypoints and match them using the Offline and Online GTs against the known landmarks.
3) Estimate the camera pose from these correspondences using a P3P algorithm and RANSAC [17]. The estimated pose is refined via a non-linear optimization.
4) Refine the location estimates of the inlier landmarks using an Extended Kalman filter.
5) Create new landmarks. Choose a number of detected keypoints that *do not* belong to any landmark in the map and initialize the new landmarks with a large uncertainty along the line of sight and a much smaller uncertainty in the camera's lateral directions.
6) Retrain GTs with good matches from 2) and the new landmarks.
7) Loop to step 2.

With this system we demonstrate that both smooth tracking and recovery from complete failure can be naturally integrated by employing GTs for the matching task.

The reconstructed trajectory in Fig. 10 (f) shows only tiny jags at the order of a few millimeters and appears as smooth as a trajectory that was estimated in a filtered approach to SLAM, e.g. MonoSLAM [18,19]. This is especially noteworthy as the camera's state is re-estimated from scratch *in every frame* and there is no such thing as a motion model.[7] At the same time, this is a strong indication for an overall correct operation, since an incorrect map induces an unstable state estimation and vice versa. In total the system mapped 724 landmarks and ran stable over all 2554 frames of the sequence. A few frames are shown in Fig. 10 (b–e).

Recently, [7] presented a system that is also capable of recovering from complete failure. They achieved robustness with a hybrid combination between template matching and a modified version of Randomized Trees. However, their map typically contains one order of magnitude fewer landmarks and there has been no indication that the modified Trees will still be capable of handling a larger number of interest-points.

The system successfully passed a rough but quantitative validation step. First, we checked the relative accuracy for reconstructed pairs of 3D points and we found an error between 3.5 to 8.8% on their Euclidean distances. Second, the absolute accuracy was assessed by choosing two world planes A_1 and A_2 parallel

[6] In this context, *to train* simply means *computing the signature and adding it to the database.*

[7] Other systems commonly use a motion model to predict the feature location in the next frame and accordingly restrict the search area for template matching.

to the ground plane. They are shown in Fig. 10 (a). We then measured their z-coordinates z_k^*, $k = \{1, 2\}$, by hand and computed for each of them the RMS error

$$e_k = \left(\frac{1}{|\mathbf{R}|} \sum_{\mathbf{R}} (z_i - z_k^*)^2 \right)^{\frac{1}{2}}$$
$$\text{with } \mathbf{R} = \{(x_i, y_i, z_i) \in \mathbf{L} \mid x_i \in [x_{\min}, x_{\max}], y_i \in [y_{\min}, y_{\max}]\}.$$

$\{x, y\}_{\min,\max}$ are the plane bounds and $\mathbf{L}$ is the set of all landmarks. We found $e_1 = 7$ mm and $e_2 = 10$ mm. Given that the camera is at roughly 0.6 to 1.4 m from the points under consideration, this represents a good accuracy.

5 Conclusion and Future Work

We have proposed a method that combines the strengths of two fundamentally different approaches to patch recognition. On one hand, the SIFT descriptor [1] established a good reputation regarding accuracy at the expense on computation time. On the other hand, statistical learning based approaches were found to be very fast at runtime but need an offline training phase.

The Generic Trees presented in this work achieve both speed and robustness to perspective and illumination changes by computing a signature based on the response of a statistical classifier trained using a small set of keypoints. In our current implementation, this set has neither been engineered nor selected to achieve maximal performance. We therefore see a large potential for improvement by seeking to optimize our choice of base keypoints.

References

1. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 20, 91–110 (2004)
2. Mikolajczyk, K., Schmid, C.: A Performance Evaluation of Local Descriptors. In: Conference on Computer Vision and Pattern Recognition, pp. 257–263 (2003)
3. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. International Journal of Computer Vision 65, 43–72 (2005)
4. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: European Conference on Computer Vision (2006)
5. Lepetit, V., Fua, P.: Keypoint recognition using randomized trees. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 1465–1479 (2006)
6. Ozuysal, M., Fua, P., Lepetit, V.: Fast Keypoint Recognition in Ten Lines of Code. In: Conference on Computer Vision and Pattern Recognition, Minneapolis, MI (2007)
7. Williams, B., Klein, G., Reid, I.: Real-time slam relocalisation. In: International Conference on Computer Vision (2007)
8. Schmid, C., Mohr, R.: Local Grayvalue Invariants for Image Retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 530–534 (1997)

9. Sivic, J., Zisserman, A.: Video Google: Efficient visual search of videos. In: Ponce, J., Hebert, M., Schmid, C., Zisserman, A. (eds.) Toward Category-Level Object Recognition. LNCS, vol. 4170, pp. 127–144. Springer, Heidelberg (2006)
10. Nister, D., Stewenius, H.: Scalable Recognition with a Vocabulary Tree. In: Conference on Computer Vision and Pattern Recognition (2006)
11. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 594–611 (2006)
12. Amit, Y., Geman, D.: Shape Quantization and Recognition with Randomized Trees. Neural Computation 9, 1545–1588 (1997)
13. Beis, J., Lowe, D.: Shape Indexing using Approximate Nearest-Neighbour Search in High-Dimensional Spaces. In: Conference on Computer Vision and Pattern Recognition, Puerto Rico, pp. 1000–1006 (1997)
14. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: European Conference on Computer Vision (2006)
15. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM: A factored solution to the simultaneous localization and mapping problem. In: Proceedings of the AAAI National Conference on Artificial Intelligence, Edmonton, Canada. AAAI Press, Menlo Park (2002)
16. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI), Acapulco, Mexico (2003)
17. Fischler, M., Bolles, R.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications ACM 24, 381–395 (1981)
18. Davison, A.J.: Real-Time Simultaneous Localisation and Mapping with a Single Camera. ICCV 02, 1403 (2003)
19. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: Monoslam: Real-time single camera slam. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 1052–1067 (2007)

Active Matching

Margarita Chli and Andrew J. Davison

Imperial College London, London SW7 2AZ, UK
{mchli,ajd}@doc.ic.ac.uk

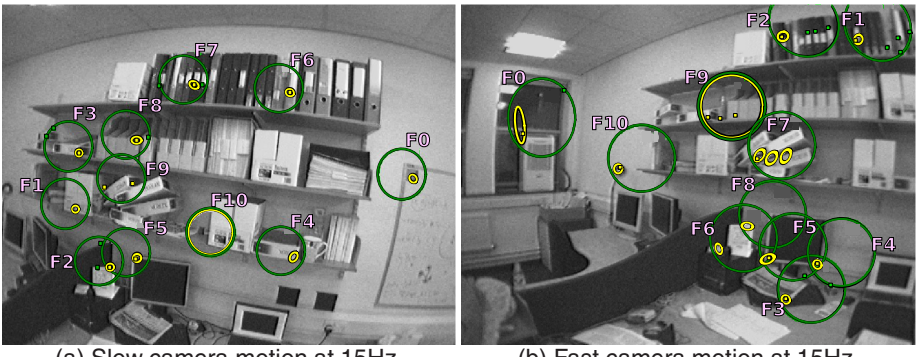
Abstract. In the matching tasks which form an integral part of all types of tracking and geometrical vision, there are invariably priors available on the absolute and/or relative image locations of features of interest. Usually, these priors are used post-hoc in the process of resolving feature matches and obtaining final scene estimates, via 'first get candidate matches, then resolve' consensus algorithms such as RANSAC. In this paper we show that the dramatically different approach of using priors dynamically to guide a feature by feature matching search can achieve global matching with much fewer image processing operations and lower overall computational cost. Essentially, we put image processing *into the loop* of the search for global consensus. In particular, our approach is able to cope with significant image ambiguity thanks to a dynamic mixture of Gaussians treatment. In our fully Bayesian algorithm, the choice of the most efficient search action at each step is guided intuitively and rigorously by expected Shannon information gain. We demonstrate the algorithm in feature matching as part of a sequential SLAM system for 3D camera tracking. Robust, real-time matching can be achieved even in the previously unmanageable case of jerky, rapid motion necessitating weak motion modelling and large search regions.

1 Introduction

It is well known that the key to obtaining correct feature associations in potentially ambiguous image matching tasks is to search for a set of correspondences which are in *consensus*: they are all consistent with a believable global hypothesis. The usual approach taken to search for matching consensus is as follows: first candidate matches are generated, for instance by detecting all features in both images and pairing features which are nearby in image space and have similar appearance. Then, incorrect 'outlier' matches are pruned by proposing and testing hypotheses of global parameters which describe the world state of interest — the 3D position of an object or the camera itself, for instance. The sampling and voting algorithm RANSAC [6] has been widely used to achieve this in geometrical vision problems.

Outliers are match candidates which lie outside of bounds determined by global consensus constraints: these are priors on the true absolute and relative locations of features if expressed in a proper probabilistic manner. The idea that inevitable outlier matches must be 'rejected' from a large number of candidates achieved by some blanket initial image processing is deeply entrenched in computer vision. The approach in the active matching paradigm of this paper is very different — to cut outliers out at source wherever possible by searching only the parts of the image where true positive matches are most probable. Instead of searching for all features and then resolving, feature searches

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 72–85, 2008.



(a) Slow camera motion at 15Hz (b) Fast camera motion at 15Hz

Fig. 1. Active matching dramatically reduces image processing operations while still achieving global matching consensus. Here, in a search for 11 point features in 3D camera tracking we contrast green regions for standard feature search with the much smaller yellow ellipses searched by our Active Matching method. In these frames, joint compatibility needed to search a factor of 4.8 more image area than Active Matching in (a) and a factor or 8.4 in (b). Moreover, JCBB encounters all the matches shown (blobs), whereas Active Matching only finds the yellow blobs.

occur one by one. The results of each search, via an exhaustive but concentrated template checking scan within a region, affect the regions within which it is likely that each of the other features will lie. This is thanks to the same inter-feature correlations of which standard consensus algorithms take advantage — but our algorithm's dynamic updating of these regions within the matching search itself means that low probability parts of the image are *never examined at all* (see Figure 1), and the number of image processing operations required to achieve global matching is reduced by a large factor. Information theory intelligently guides the step by step search process from one search region to the next and can even indicate when matching should be terminated at a point of diminishing returns.

While matching is often formulated as a search for correspondence between one image and another (for example in the literature on 3D multi-view constraints with concepts such as the multi-view tensors), stronger constraints are available when we consider matching an image to a *state* — an estimate of world properties perhaps accumulated over many images. Uncertainty in a state is represented with a probability distribution. Matching constraints are obtained by projecting the uncertain world state into a new image, the general result being a joint prior probability distribution over the image locations of features. These uncertain feature *predictions* will often be highly correlated. When probabilistic priors are available, the unsatisfying random sampling and preset thresholds of RANSAC have been improved on by probabilistic methods such as the Joint Compatibility Branch and Bound (JCBB) algorithm [11] which matches features via a deterministic interpretation tree [7] and has been applied to geometric image matching in [1]. JCBB takes account of a joint Gaussian prior on feature positions and calculates the joint probability that any particular hypothesized set of correspondences is correct.

Our algorithm aims to perform at least as well as JCBB in determining global consensus while searching much smaller regions of an image. It goes much further than

previously published 'guided matching' algorithms such as [12] in guiding not just a search for consensus but the image processing to determine candidate matches themselves.

Davison [3] presented a theoretical analysis of information gain in sequential image search. However, this work had the serious limitation of representing the current estimate of the state of the search at all times with a single multi-variate Gaussian distribution. This meant that while theoretically and intuitively satisfying active search procedures were demonstrated in simulated problems, the technique was not applicable to real image search because of the lack of ability to deal with discrete multiple hypotheses which arise due to matching ambiguity — only simulation results were given. Here we use a dynamic mixture of Gaussians (MOG) representation which grows as necessary to represent the discrete multiple hypotheses arising during active search. We show that this representation can now be applied to achieve highly efficient image search in real, ambiguous tracking problems.

2 Probabilistic Prediction and Feature by Feature Search

We consider making image measurements of an object or scene of which the current state of knowledge is modelled by a probability distribution over a finite vector of parameters $\mathbf{x}$ — representing the position of a moving object or camera, for instance. In an image, we are able to observe *features*: measurable projections of the scene state. A measurement of feature i yields the vector of parameters $\mathbf{z}_i$ — for example the 2D image coordinates of a keypoint. A likelihood function $p(\mathbf{z}_i|\mathbf{x})$ models the measurement process.

When a new image arrives, we can project the current probability distribution over state parameters $\mathbf{x}$ into feature space to *predict* the image locations of all the features which are measurement candidates. Defining stacked vector $\mathbf{z}_T = \begin{pmatrix} \mathbf{z}_1 & \mathbf{z}_2 & \dots \end{pmatrix}^\top$ containing all candidate feature measurements, the density:

$$p(\mathbf{z}_T) = \int p(\mathbf{z}_T|\mathbf{x})p(\mathbf{x})d\mathbf{x} \; . \tag{1}$$

is a probabilistic prediction not just of the most likely image position of each feature, but a joint distribution over the expected locations of all of them. Given just individually marginalised parts $p(\mathbf{z}_i)$ of this prediction, the image search for each feature can sensibly be limited to high-probability regions, which will practically often be small in situations such as tracking. In Isard and Blake's Condensation [8], for example, feature searches take place in fixed-size windows around pre-determined measurement sites centred at a projection into measurement space of each of the particles representing the state probability distribution.

However, the extra information available that has usually been overlooked in feature search but which we exploit in this paper is that the predictions of the values of all the candidate measurements which make up joint vector $\mathbf{z}_T$ are often highly correlated, since they all depend on common parts of the scene state $\mathbf{x}$. In a nutshell, the correlation between candidate measurements means that making a measurement of one feature tells us a lot about where to look for another feature, suggesting a step by step guided search rather than blanket examination of all feature regions.

2.1 Guiding Search Using Information Theory

At each step in the search, the next feature and search region must be selected. Such candidate measurements vary in two significant ways: the amount of information which they are expected to offer, and the amount of image processing likely to be required to extract a match; both of these quantities can be computed directly from the current search prior. There are ad-hoc ways to score the value of a measurement such as search ellipse size, used for simple active search for instance in [5]. However, Davison [3], building on early work by others such as Manyika [10], explained clearly that the Mutual Information (MI) between a candidate and the scene state is the essential probabilistic measure of measurement value.

Following the notation of Mackay [9], the (MI) of continuous multivariate PDFs $p(\mathbf{x})$ and $p(\mathbf{z}_i)$ is:

$$I(\mathbf{x};\mathbf{z}_i) = E\left[\log_2 \frac{p(\mathbf{x}|\mathbf{z}_i)}{p(\mathbf{x})}\right] \tag{2}$$

$$= \int_{\mathbf{x},\mathbf{z}_i} p(\mathbf{x},\mathbf{z}_i) \log_2 \frac{p(\mathbf{x}|\mathbf{z}_i)}{p(\mathbf{x})} d\mathbf{x} d\mathbf{z}_i \ . \tag{3}$$

Mutual information is *expected information gain*: $I(\mathbf{x};\mathbf{z}_i)$ is how many **bits** of information we expect to learn about the uncertain vector $\mathbf{x}$ by determining the exact value of $\mathbf{z}_i$. In active matching, the MI scores of the various candidate measurements $\mathbf{z}_i$ can be fairly compared to determine which has most utility in reducing uncertainty in the state $\mathbf{x}$, even if the measurements are of different types (e.g. point feature vs. edge feature). Further, dividing MI by the computational cost required to extract a measurement leads to an 'information efficiency' score [3] representing the bits to be gained per unit of computation.

We also see here that when evaluating candidate measurements, a useful alternative to calculating the mutual information $I(\mathbf{x};\mathbf{z}_i)$ between a candidate measurement and the state is to use the MI $I(\mathbf{z}_{T\neq i};\mathbf{z}_i)$ between the candidate and *all the other candidate measurements*. This is a measure of how much information the candidate would provide about the other candidates, capturing the core aim of an active search strategy to decide on measurement order. This formulation has the very satisfying property that active search can proceed purely in measurement space, and is appealing in problems where it is not desirable to make manipulations of the full state distribution during active search.

2.2 Active Search Using a Single Gaussian Model

To attack the coupled search problem, Davison [3] made the simplifying assumption that the PDFs describing knowledge of $\mathbf{x}$ and $\mathbf{z}_T$ can be approximated always by single multi-variate Gaussian distributions. The measurement process is modelled by $\mathbf{z}_i = \mathbf{h}_i(\mathbf{x}) + \mathbf{n}_m$, where $\mathbf{h}_i(\mathbf{x})$ describes the functional relationship between the expected measurement and the object state as far as understood via the models used of the object and sensor, and $\mathbf{n}_m$ is a Gaussian-distributed vector representing unmodelled effects (noise) with covariance $\mathtt{R}_i$ which is independent for each measurement. The vector $\mathbf{x}_m$ which stacks the object state and candidate measurements (in measurement space) can be calculated along with its full covariance:

$$\hat{\mathbf{X}}_m = \begin{pmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{z}}_1 \\ \hat{\mathbf{z}}_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{x}} \\ \mathbf{h}_1(\hat{\mathbf{x}}) \\ \mathbf{h}_2(\hat{\mathbf{x}}) \\ \vdots \end{pmatrix}, \; \mathtt{P}_{\mathbf{x}_m} = \begin{bmatrix} \mathtt{P}_x & \mathtt{P}_x \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}}^\top & \mathtt{P}_x \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}}^\top & \cdots \\ \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}} \mathtt{P}_x & \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}} \mathtt{P}_x \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}}^\top + \mathtt{R}_1 & \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}} \mathtt{P}_x \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}}^\top & \cdots \\ \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}} \mathtt{P}_x & \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}} \mathtt{P}_x \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}}^\top & \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}} \mathtt{P}_x \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}}^\top + \mathtt{R}_2 & \cdots \\ \vdots & \vdots & \vdots & \end{bmatrix} \quad (4)$$

The lower-right portion of $\mathtt{P}_{\mathbf{x}_m}$ representing the covariance of $\mathbf{z}_T = \begin{pmatrix} \mathbf{z}_1 \; \mathbf{z}_2 \; \ldots \end{pmatrix}^\top$ is known as the *innovation covariance matrix* $\mathtt{S}$ in Kalman filter tracking. The correlations between different candidate measurements mean that generally $\mathtt{S}$ will not be block-diagonal but contain off-diagonal correlations between the predicted measurements of different features.

With this single Gaussian formulation, the mutual information in bits between any two partitions α and β of $\mathbf{x}_m$ can be calculated according to this formula:

$$I(\alpha;\beta) = \frac{1}{2}\log_2 \frac{|\mathtt{P}_{\alpha\alpha}|}{|\mathtt{P}_{\alpha\alpha} - \mathtt{P}_{\alpha\beta}\mathtt{P}_{\beta\beta}^{-1}\mathtt{P}_{\beta\alpha}|}, \quad (5)$$

where $\mathtt{P}_{\alpha\alpha}$, $\mathtt{P}_{\alpha\beta}$, $\mathtt{P}_{\beta\beta}$ and $\mathtt{P}_{\beta\alpha}$ are sub-blocks of $\mathtt{P}_{\mathbf{x}_m}$. This representation however can be computationally expensive as it involves matrix inversion and multiplication so exploiting the properties of mutual information we can reformulate into:

$$I(\alpha;\beta) = H(\alpha) - H(\alpha|\beta) = H(\alpha) + H(\beta) - H(\alpha,\beta) \quad (6)$$

$$= \frac{1}{2}\log_2 \frac{|\mathtt{P}_{\alpha\alpha}||\mathtt{P}_{\beta\beta}|}{|\mathtt{P}_{\mathbf{x}_m}|}. \quad (7)$$

2.3 Multiple Hypothesis Active Search

The weakness of the single Gaussian approach of the previous section is that, as ever, a Gaussian is uni-modal and can only represent a PDF with one peak. In real image search problems no match (or failed match) can be fully trusted: true matches are sometimes missed (false negatives), and clutter similar in appearance to the feature of interest can lead to false positives. This is the motivation for the mixture of Gaussians formulation used in our active matching algorithm. We wish to retain the feature-by-feature quality of active search. The MOG representation allows dynamic, online updating of the multi-peaked PDF over feature locations which represents the multiple hypotheses which arise during as features are matched ambiguously.

3 Active Matching Algorithm

Our active matching algorithm searches for global correspondence in a series of steps which gradually refine the probabilistic 'search state' initially set as the prior on feature positions. Each step consists of a search for a template match to one feature within a certain bounded image region, followed by an update of the search state which depends on the search outcome. After many well-chosen steps the search state collapses to a highly peaked posterior estimate of image feature locations — and matching is finished.

3.1 Search State Mixture of Gaussians Model

A single multi-variate Gaussian probability distribution over vector $\mathbf{x}_m$ defined in Equation 4 is parameterised by a 'mean vector' $\hat{\mathbf{x}}_m$ and covariance matrix $\mathtt{P}_{\mathbf{x}_m}$, and we use the shorthand $\mathbf{G}(\hat{\mathbf{x}}_m, \mathtt{P}_{\mathbf{x}_m})$ to represent the explicit normalised PDF:

$$p(\mathbf{x}_m) = \mathbf{G}(\hat{\mathbf{x}}_m, \mathtt{P}_{\mathbf{x}_m}) \tag{8}$$

$$= (2\pi)^{-\frac{D}{2}} |\mathtt{P}_{\mathbf{x}_m}|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}_m - \hat{\mathbf{x}}_m)^\top \mathtt{P}_{\mathbf{x}_m}^{-1} (\mathbf{x}_m - \hat{\mathbf{x}}_m)} . \tag{9}$$

During active matching, we now represent the PDF over $\mathbf{x}_m$ with a multi-variate MOG distribution formed by the sum of K individual Gaussians each with weight λ_i:

$$p(\mathbf{x}) = \sum_{i=1}^{K} p(\mathbf{x}_i) = \sum_{i=1}^{K} \lambda_i \mathbf{G}_i , \tag{10}$$

where we have now used the further notational shorthand $\mathbf{G}_i = \mathbf{G}(\hat{\mathbf{x}}_{m_i}, \mathtt{P}_{\mathbf{x}_{m_i}})$. Each Gaussian distribution must have the same dimensionality. We normally assume that the input prior at the start of the search process is well-represented by a single Gaussian and therefore $\lambda_1 = 1, \lambda_{i \neq 1} = 0$. As active search progresses and there is a need to propagate multiple hypotheses, this and subsequent Gaussians will divide as necessary, so that at a general instant there will be K Gaussians with normalised weights $\sum_{i=1}^{K} \lambda_i = 1$.

The current MOG search state model forms the prior for a step of active matching. This prior, and the likelihood and posterior distributions to be explained in the following sections, are shown in symbolic 1D form in Section 3.4.

3.2 The Algorithm

The active matching algorithm (see Figure 2) is initialized with a joint Gaussian prior over the features' locations in measurement space (e.g. prediction after application of motion model). At each step we select a {Gaussian, Feature} pair for measurement based on the expected information gain (see Section 4) and make an exhaustive search for feature matches within this region, finding zero or more matches above a threshold. For every template match yielded by the search a new Gaussian is spawned with mean and covariance conditioned on the hypothesis of that match being a true positive, and we also consider the 'null' possibility that none of the matches is a true positive. After a search the MoG distribution is updated to represent the outcome, as detailed in the rest of this section. Very weak Gaussians are pruned from the mixture after each search step. The algorithm continues until all features have been measured, or an alternative stopping criterion can be defined based on expected information gain falling below a desired value indicating that nothing more of relevance is to be obtained from the image.

3.3 Likelihood Function

One step of active matching takes place by searching the region defined by the high-probability 3σ extent of one of the Gaussians in the measurement space of the selected feature. If we find M candidate template matches and no match elsewhere $\mathbf{z}_c = \{\mathbf{z}_1 \ldots \mathbf{z}_M \mathbf{z}'_{rest}\}$ then the likelihood $p(\mathbf{z}_c|\mathbf{x})$ of this result is modelled as a mixture:

ACTIVEMATCHING($\mathbf{G}_{in}$)	UPDATEMIXTURE(Mixture$_{1..K}$,Matches$_{1..M}$)
Mixture = $[[1, \mathbf{G}_{in}]]$	$[\lambda_i, \mathbf{G}_i]$ = get_measured_gaussian(Mixture)
$\{\mathbf{F}_{sel}, \mathbf{G}_{sel}\}$ = get_max_gain_pair(Mixture)	**for** $m = 1 : M$
while is_unmeasured($\mathbf{F}_{sel}, \mathbf{G}_{sel}$)	$[\lambda_m, \mathbf{G}_m]$ = fuse_match($\mathbf{G}_i$,λ_i,Matches[m])
Matches = measure($\mathbf{F}_{sel}, \mathbf{G}_{sel}$)	Mixture = [Mixture,$[\lambda_m, \mathbf{G}_m]$]
UPDATEMIXTURE(Mixture,Matches)	**end for**
prune_weak_gaussians(Mixture)	**for** $k = 1 : K$
$\{\mathbf{F}_{sel}, \mathbf{G}_{sel}\}$ = get_max_gain_pair(Mixture)	$\lambda_{k,new}$ = update_weight(λ_k,Matches)
end while	Mixture[k] = $[\lambda_{k,new}, \mathbf{G}_k]$
$\mathbf{G}_{best}$ = get_most_probable_gaussian(Mixture)	**end for**
return $\mathbf{G}_{best}$	normalize_weights(Mixture)
	return

Fig. 2. Active matching algorithm and UPDATEMIXTURE sub-procedure

M Gaussians $\mathbf{H_m}$ representing the hypotheses that each candidate is the true **match** (these Gaussians, functions of $\mathbf{x}$, having the width of the measurement uncertainty $\mathtt{R}_i$), and two constant terms representing the hypotheses that the candidates are all spurious false positives and the true match lies either **in** or **out** of the searched region:

$$p(\mathbf{z}_c|\mathbf{x}) = \mu_{\text{in}}\mathbf{T}_{\text{in}} + \mu_{\text{out}}\mathbf{T}_{\text{out}} + \sum_{\mathbf{m=1}}^{M} \mu_{\text{match}}\mathbf{H_m} \,. \tag{11}$$

If N is the total number of pixels in the search region, then the constants in this expression have the form:

$$\mu_{\text{in}} = P_{\mathbf{fp}}^{M} P_{\mathbf{fn}} P_{\mathbf{tn}}^{N-(M+1)} \tag{12}$$

$$\mu_{\text{out}} = P_{\mathbf{fp}}^{M} P_{\mathbf{tn}}^{N-M} \tag{13}$$

$$\mu_{\text{match}} = P_{\mathbf{tp}} P_{\mathbf{fp}}^{M-1} P_{\mathbf{tn}}^{N-M} \,, \tag{14}$$

where $P_{\mathbf{tp}}$, $P_{\mathbf{fp}}$ $P_{\mathbf{tn}}$, $P_{\mathbf{fn}}$ are per-pixel true positive, false positive, true negative and false negative probabilities respectively for the feature. $\mathbf{T}_{\text{in}}$ and $\mathbf{T}_{\text{out}}$ are top-hat functions with value one inside and outside of the searched Gaussian $\mathbf{H_m}$ respectively and zero elsewhere, since the probability of a null search depends on whether the feature is really within the search region or not. Given that there can only be one true match in the searched region, μ_{in} represents the probability that we record M false positives, one false negative and $N - (M + 1)$ true negatives. μ_{out} represents the probability of M false positives and $N - M$ true negatives. The μ_{match} weight of a Gaussian hypothesis of a true match represents one true positive, $M - 1$ false positives and $N - M$ true negatives.

3.4 Posterior: Updating After a Measurement

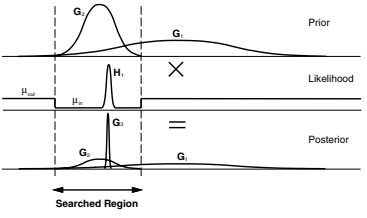
The standard application of Bayes' Rule to obtain the posterior distribution for $\mathbf{x}$ given the new measurement is:

$$p(\mathbf{x}|\mathbf{z}_c) = \frac{p(\mathbf{z}_c|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z}_c)} \,. \tag{15}$$

Substituting MOG models from Equations 10 and 11:

$$p(\mathbf{x}|\mathbf{z}_c) = \frac{\left(\mu_{\text{in}}\mathbf{T}_{\text{in}} + \mu_{\text{out}}\mathbf{T}_{\text{out}} + \sum_{\mathbf{m=1}}^{M} \mu_{\text{match}}\mathbf{H}_{\mathbf{m}}\right)\left(\sum_{i=1}^{K} \lambda_i \mathbf{G}_i\right)}{p(\mathbf{z}_c)} . \quad (16)$$

The denominator $p(\mathbf{z}_c)$ is a constant determined by normalising all new weights λ_i to add up to one). In the illustration below illustrating the formation of a posterior, we show an example of $M = 1$ match. This posterior will then become the prior for the next active matching step.



In the top line of Equation 16, the product of the two MOG sums will lead to K scaled versions of all the original Gaussians and MK terms which are the products of two Gaussians. However, we make the approximation that only M of these MK Gaussian product terms are significant: those involving the prior Gaussian currently being measured. We assume that since the other Gaussians in the prior distribution are either widely separated or have very different weights, the resulting products will be negligible. Therefore there are only M new Gaussians added to the mixture: generally highly-weighted, spiked Gaussians corresponding to new matches in the searched region. These are considered to be '*children*' of the searched parent Gaussian. An important point to note is that if multiple matches in a search region lead to several new child Gaussians being added, one corresponding to a match close to the centre of the search region will correctly have a higher weight than others, having been formed by the product of a prior and a measurement Gaussian with nearby means.

All other existing Gaussians get updated posterior weights by multiplication with the constant terms. Note that the null information of making a search where no template match is found is fully accounted for in our framework — in this case we will have $M = 0$ and no new Gaussians will be generated, but the weight of the searched Gaussian will diminish.

Finally, very weak Gaussians (with weight < 0.001) are pruned from the mixture after each search step. This avoids otherwise rapid growth in the number of Gaussians such that in practical cases fewer than 10 Gaussians are 'live' at any point, and most of the time far fewer than this. This pruning is the better, fully probabilistic equivalent in the dynamic MOG scheme of lopping off branches in an explicit interpretation tree search such as JCBB [11].

4 Measurement Selection

4.1 Search Candidates

At each step of the MOG active matching process, we use the mixture $p(\mathbf{x}_m)$ to predict individual feature measurements, and there are KF possible actions, where K is the

number of Gaussians and F is the number of measurable features. We rule out any $\{$Gaussian, Feature$\}$ combinations where we have already made a search. Also ruled out are 'child' Gaussians for a certain feature which lie completely within an already searched ellipse. For example, if we have measured root Gaussian $\mathbf{G}_1$ at feature 1, leading to the spawning of $\mathbf{G}_2$ which we search at feature 3 to spawn $\mathbf{G}_3$, then the candidates marked with '$*$' would be ruled out from selection:

	$\mathbf{F}_1$	$\mathbf{F}_2$	$\mathbf{F}_3$	$\mathbf{F}_4$
$\mathbf{G}_1$	$*$			

$\Rightarrow$

	$\mathbf{F}_1$	$\mathbf{F}_2$	$\mathbf{F}_3$	$\mathbf{F}_4$
$\mathbf{G}_1$	$*$			
$\mathbf{G}_2$	$*$		$*$	

$\Rightarrow$

	$\mathbf{F}_1$	$\mathbf{F}_2$	$\mathbf{F}_3$	$\mathbf{F}_4$
$\mathbf{G}_1$	$*$			
$\mathbf{G}_2$	$*$		$*$	
$\mathbf{G}_3$	$*$		$*$	

All of the remaining candidates are evaluated in terms of mutual information with the state or other candidate measurements, and then selected based on an information efficiency score [3] which is this mutual information divided by the area of the search region, assumed proportional to search cost.

4.2 Mutual Information for a Mixture of Gaussians Distribution

In order to assess the amount of information that each candidate $\{$Feature, Gaussian$\}$ measurement pair can provide, we predict the post-search mixture of Gaussians depending on the possible outcome of the measurement: (1): A **null search**, where no template match is found above a threshold. The effect is only to change the weights of the current Gaussians in the mixture into λ_i'. (2): A **template match**, causing a new Gaussian to be spawned with reduced width as well as re-distributing the weights of the all Gaussians of the new mixture to λ_i''.

In a well-justified assumption of 'weakly-interacting Gaussians' which are either well-separated or have dramatically different weights, we separate the information impact of each candidate measurement into two components: (a) I_{discrete} captures the effect of the redistribution of weights depending on the search outcome (the desire to *reduce ambiguity*), which (b) $I_{\text{continuous}}$ gives a measure of the reduction in covariance of the most likely Gaussian which becomes more peaked after a match (the desire to *increase precision*). Due to the intuitive absolute nature of mutual information, these terms are additive:

$$I = I_{\text{discrete}} + I_{\text{continuous}} \tag{17}$$

One of other of these terms will dominate at different stages of the matching process, depending on whether the key uncertainty is due to discrete ambiguity or continuous accuracy. It is highly appealing that this behaviour arises automatically thanks to the MI formulation.

Mutual Information: Discrete Component. Considering the effect of a candidate measurement purely in terms of the change in the weights of the Gaussians in the mixture, we calculate the mutual information it is predicted to provide by:

$$I(\mathbf{x};\mathbf{z}) = H(\mathbf{x}) - H(\mathbf{x}|\mathbf{z}). \tag{18}$$

Given that the search outcome can have two possible states (null or match-search), then:

$$I_{\text{discrete}} = H(\mathbf{x}) - P(\mathbf{z} = \text{null}) \quad H(\mathbf{x}|\mathbf{z} = \text{null}) \tag{19}$$

$$- P(\mathbf{z} = \text{match}) H(\mathbf{x}|\mathbf{z} = \text{match}) \,. \tag{20}$$

where

$$H(\mathbf{x}) = \sum_{i=1}^{K} \lambda_i \log_2 \frac{1}{\lambda_i}, \; H(\mathbf{x}|\mathbf{z} = \text{null}) = \sum_{i=1}^{K} \lambda'_i \log_2 \frac{1}{\lambda'_i}, \; H(\mathbf{x}|\mathbf{z} = \text{match}) = \sum_{i=1}^{K+1} \lambda''_i \log_2 \frac{1}{\lambda''_i}. \tag{21}$$

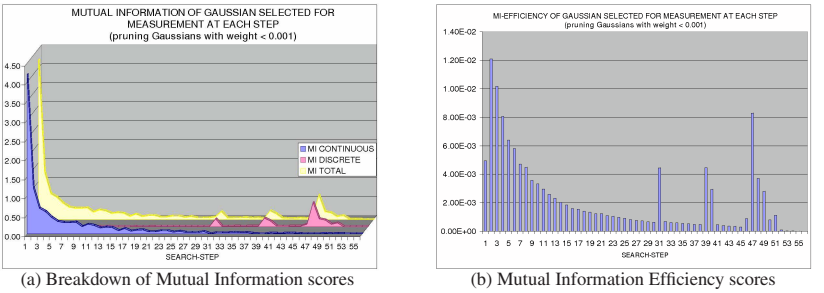
The predicted weights after a null or a match search are calculated as in Equation 16 with the only difference that the likelihood of a match-search is summed over all positions in the search-region that can possibly yield a match.

Mutual Information: Continuous Component. Continuous MI is computed using Equation 7:

$$I_{\text{continuous}} = \frac{1}{2} P(\mathbf{z} = \text{match}) \lambda''_m \log_2 \frac{|\mathrm{P}_{\alpha\alpha}||\mathrm{P}_{\beta\beta}|}{|\mathrm{P}_{\mathbf{x}_m}|} \tag{22}$$

This captures the information gain associated with the shrinkage of the measured Gaussian (λ''_m is the predicted weight of the new Gaussian evolving) thanks to the positive match: if the new Gaussian has half the determinant of the old one, that is one bit of information gain. This was the only MI term considered in [3] but is now scaled and combined with discrete component arising due to the expected change in the λ_i distribution. As explained in Section 2, we can replace the product $|\mathrm{P}_{\alpha\alpha}||\mathrm{P}_{\beta\beta}|$ with $|\mathrm{P}_{\mathbf{z}_{T\neq i}}||\mathrm{P}_{\mathbf{z}_{T=i}}|$ to calculate a continuous MI score in measurement space.

Figure 3(a, b) shows the MI and MI efficiency scores of the selected measurement at each step of the matching process when Active Matching is applied to a frame from MonoSLAM (see Section 5) with around 50 candidate features. These plots demonstrate the expected tailing off of measurement utility and the diminishing returns of continued search.



(a) Breakdown of Mutual Information scores

(b) Mutual Information Efficiency scores

Fig. 3. The evolution of MI and MI-efficiency scores of the selected measurement through the search-steps of Active Matching within a frame, tracking on average 50 features. Both values tail off generally with spikes as null searches or ambiguities arise and send search in a different direction. In (a) the total Mutual Information is shown broken down into its discrete and continuous components. It is the continuous component which displays a smooth decay with search step number, while the discrete component spikes up at ambiguities.

5 Results

We present results on the application of the algorithm to feature matching within the publically available MonoSLAM system [4] for real-time probabilistic structure and motion estimation. This system, which is well known for its computational efficiency thanks to predictive search, uses an Extended Kalman Filter to estimate the joint distribution over the 3D location of a calibrated camera and a sparse set of point features — here we use it to track the motion of a hand-held camera in an office scene with image capture at 15 or 30Hz. At each image of the real-time sequence, MonoSLAM applies a probabilistic motion model to the accurate posterior estimate of the previous frame, adding uncertainty to the camera part of the state distribution. In standard configuration it then makes independent probabilistic predictions of the image location of each of the features of interest, and each feature is independently searched for by an exhaustive template matching search within the ellipse defined by a three standard deviation gate. The top-scoring template match is taken as correct if its normalised SSD score passes a threshold. At low levels of motion model uncertainty, mismatches via this method are relatively rare, but in advanced applications of the algorithm [1,13] it has been observed that Joint Compatibility testing finds a significant number of matching errors and greatly improves performance.

Our active matching algorithm simply takes as input from MonoSLAM the predicted stacked measurement vector $\mathbf{z}_T$ and innovation covariance matrix $\mathtt{S}$ and returns a list of globally matched feature locations. We have implemented a straightforward feature statistics capability within MonoSLAM to sequentially record the average number of locations in an image similar to each of the mapped features, counting successful and failed match attempts in the feature's true location. This is used to assess false positive and false negative rates for each feature. More sophisticated online methods for assessing feature statistics during mapping have recently been published [2]. An example of how ambiguity is handled and resolved by active matching within a typical MonoSLAM frame is shown in Figure 4.

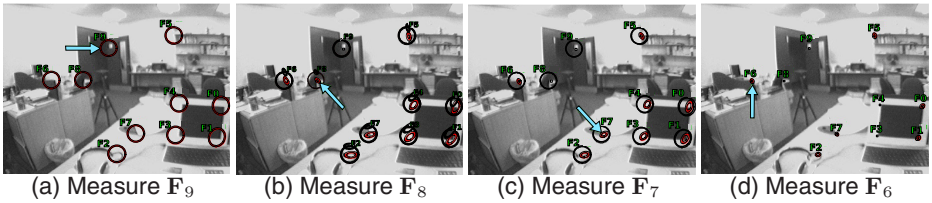

(a) Measure $\mathbf{F}_9$ (b) Measure $\mathbf{F}_8$ (c) Measure $\mathbf{F}_7$ (d) Measure $\mathbf{F}_6$

Fig. 4. Resolving ambiguity in MonoSLAM using active matching. Starting from (a) showing single Gaussian $\mathbf{G}_0$ set to the image prior at the start of matching, red ellipses represent the most probable Gaussian at each step and the arrows denote the {Feature,Gaussian} combination selected for measurement guided by MI efficiency. Feature 9 yields 2 matches and therefore two new Gaussians evolve in (b), $\mathbf{G}_1$ (small red) and $\mathbf{G}_2$(small black) . Successful measurement of Feature 8 in $\mathbf{G}_1$ lowers the weight of $\mathbf{G}_2$ (0.00013) so in (c) it gets pruned from the mixture. Despite the unsuccessful measurement of Feature 7 in $\mathbf{G}_1$, after successful measurements of Features 3 and 4, there is only one Gaussian left in the mixture, with very small search-regions for all yet-unmeasured features.

5.1 Sequence Results

Two different hand-held camera motions were used to capture image sequences at 30Hz: one with a standard level of dynamics slightly faster than in the results of [4], and one with much faster, jerky motion (see the video submitted with this paper). MonoSLAM's motion model parameters were tuned such that prediction search regions were wide enough that features did not 'jump out' at any point — necessitating a large process noise covariance and very large search regions for the fast sequence. Two more sequences were generated by subsampling each of the 30Hz sequences by a factor of two. These four sequences were all processed using active matching and also the combination of full searches of all ellipses standard in MonoSLAM with JCBB to prune outliers. In terms of accuracy, active matching was found to determine the same set of feature associations as JCBB on all frames of the sequences. The key difference was in the computational requirements of the algorithms, as shown below:

	One tracking step	Matching only	No. pixels searched	Max no. live Gaussians
	Fast Sequence at 30Hz (752 frames)			
JCBB	56.8ms	51.2ms	40341	
Active Matching	21.6ms	16.1ms	5039	7
	Fast Sequence at 15Hz (376 frames)			
JCBB	102.6ms	97.1ms	78675	
Active Matching	38.1ms	30.4ms	9508	10
	Slow Sequence at 30Hz (592 frames)			
JCBB	34.9ms	28.7ms	21517	
Active Matching	19.5ms	16.1ms	3124	5
	Slow Sequence at 15Hz (296 frames)			
JCBB	59.4ms	52.4ms	40548	
Active Matching	22.0ms	15.6ms	5212	6

The key result here is the ability of active matching to cope efficiently with global consensus matching at real-time speeds (looking at the 'One tracking step' total processing time column in the table) even for the very jerky camera motion which is beyond the real-time capability of the standard 'search all ellipses and resolve with JCBB' approach whose processing times exceed real-time constraints. This computational gain is due to the large reductions in the average number of template matching operations per frame carried out during feature search, as highlighted in the 'No. pixels searched' column — global consensus matching has been achieved by analysing around one eighth of the image locations needed by standard techniques. This is illustrated dramatically in Figure 1, where the regions of pixels actually searched by the two techniques are overlaid on frames from two of the sequences.

This new real-time ability to tracking extremely rapid camera motion at a range of frame-rates significantly expands the potential applications of 3D camera tracking. Please see the submitted videos for full illustration of the operation of active matching on these sequences.

5.2 Computational Complexity

We have seen that active matching will always reduce the number of image processing operations required when compared to blanket matching schemes, but it requires extra computation in calculating *where to search* at each step of the matching process. The sequence results indicate that these extra computations are more than cancelled out by the gain in image processing speed, but it is appropriate to analyse of their computational complexity.

Each step of the active matching algorithm first requires MI efficiency scores to be generated and compared for up to the KF measurable combinations of feature and current live Gaussians.

Each MI evaluation requires computation of order $O(K)$ for the discrete component and $O(F^3)$ for the continuous component using formula Equation 22 (the deterimnants can be computed by LU decomposition or similar). The constants of proportionality are small here and these evaluations are cheap for low numbers of feature candidates. Although the the cost of evaluating continuous MI scales poorly with the number of feature candidates, in practice if the image feature density is high then it will be sensible to limit the number of candidates selected between at each step: for instance one candidate could be randomly chosen from each block of a regular grid overlaid on the image, on the assumption that candidates within a small region are highly correlated and choosing between them is unnecessary.

The number of steps required to achieve global matching of all features will be around $\bar{K}F$, where $\bar{K}$ is the average number of live Gaussians after pruning. However, in practical applications with large numbers of features we will be able to improve on this by terminating the matching process when the expected information gain from any remaining candidates drops below a threshold — again, when the feature density is very high, there will be many highly correlated feature candidates and the mutual information criterion will tell us that there is little point in measuring all of them.

6 Conclusions

We have shown that a mixture of Gaussians formulation allows global consensus feature matching to proceed in a fully sequential, Bayesian algorithm which we call active matching. Information theory plays a key role in guiding highly efficient image search and we can achieve large factors in the reduction of image processing operations.

We plan to experiment with this algorithm in a range of different scenarios to gauge the effectiveness of active search at different frame-rates, resolutions, feature densities and tracking dynamics. While our initial instinct was that the algorithm would be most powerful in matching problems with strong priors such as high frame-rate tracking due to the advantage it can take of good predictions, our experiments with lower frame-rates indicate its potential also in other problems such as recognition. There priors on absolute feature locations will be weak but priors on relative locations may still be strong.

Acknowledgements

This research was supported by EPSRC grant GR/T24685/01. We are grateful to Ian Reid, José María Montiel, José Neira, Javier Civera and Paul Newman for useful discussions.

References

1. Clemente, L.A., Davison, A.J., Reid, I.D., Neira, J., Tardós, J.D.: Mapping large loops with a single hand-held camera. In: Proceedings of Robotics: Science and Systems (RSS) (2007)
2. Cummins, M., Newman, P.: Probabilistic appearance based navigation and loop closing. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2007)
3. Davison, A.J.: Active search for real-time vision. In: Proceedings of the International Conference on Computer Vision (ICCV) (2005)
4. Davison, A.J., Molton, N.D., Reid, I.D., Stasse, O.: MonoSLAM: Real-time single camera SLAM. Transactions on Pattern Analysis and Machine Intelligence (PAMI) 29(6), 1052–1067 (2007)
5. Davison, A.J., Murray, D.W.: Mobile robot localisation using active vision. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407. Springer, Heidelberg (1998)
6. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), 381–395 (1981)
7. Grimson, W.E.L.: *Object Recognition by Computer: The Role of Geometric Constraints*. MIT Press, Cambridge (1990)
8. Isard, M., Blake, A.: Contour tracking by stochastic propagation of conditional density. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1064, pp. 343–356. Springer, Heidelberg (1996)
9. Mackay, D.: *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Cambridge (2003)
10. Manyika, J.: An Information-Theoretic Approach to Data Fusion and Sensor Management. PhD thesis, University of Oxford (1993)
11. Neira, J., Tardós, J.D.: Data association in stochastic mapping using the joint compatibility test. IEEE Trans. Robotics and Automation 17(6), 890–897 (2001)
12. Tordoff, B., Murray, D.: Guided-MLESAC: Faster image transform estimation by using matching priors. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 27(10), 1523–1535 (2005)
13. Williams, B., Klein, G., Reid, I.: Real-time SLAM relocalisation. In: Proceedings of the International Conference on Computer Vision (ICCV) (2007)

Towards Scalable Dataset Construction: An Active Learning Approach

Brendan Collins, Jia Deng, Kai Li, and Li Fei-Fei

Department of Computer Science, Princeton University, New Jersey, U.S.A.
{bmcollin,dengjia,li,feifeili}@cs.princeton.edu

Abstract. As computer vision research considers more object categories and greater variation within object categories, it is clear that larger and more exhaustive datasets are necessary. However, the process of collecting such datasets is laborious and monotonous. We consider the setting in which many images have been automatically collected for a visual category (typically by automatic internet search), and we must separate relevant images from noise. We present a discriminative learning process which employs active, online learning to quickly classify many images with minimal user input. The principle advantage of this work over previous endeavors is its scalability. We demonstrate precision which is often superior to the state-of-the-art, with scalability which exceeds previous work.

1 Introduction

Though it is difficult to foresee the future of computer vision research, it is likely that its trajectory will include examining a greater number of visual categories (such as objects or scenes), that the complexity of the models employed on these categories will increase, and that these categories will include greater intraclass variation. It is unlikely that the researcher's patience for labeling images will keep pace with the growing need for annotated datasets. For this reason, this work aims to develop a system which can obtain high-precision databases of images with minimal supervision. The particular focus of this work is scalability, and its principle contribution is demonstrating the effectiveness of active learning for automatic dataset construction. With minimal supervision, we match or exceed the precision demonstrated by state-of-the-art technologies. However, using active learning, we are able to extend the performance of our system greatly as the user opportunistically labels additional images. Active learning focuses the attention of the user on those images which are most informative.

Computer vision research has always been heavily dependent on good datasets, many of which were hand-collected (e.g., Caltech-101 [1], Caltech-256 [2], PASCAL [3], LabelMe [4], Fink *et al.* [5], and LotusHill [6]). However, in the last several years, there have been a number of papers which attempt to automate this laborious task. Early work by Fergus *et al.* [7,8] re-ranked images obtained from Google Image Search using visual information.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 86–98, 2008.

Berg *et al.* [9] aim to automatically construct image datasets for several animal categories. They begin by searching the web using Google text search, obtaining images from the first 1000 web pages returned. Using Latent Dirichlet Allocation they identify a number of latent topics and corresponding exemplary images from this crawled set. These exemplary images are labeled by the user as relevant or background; this labeled set is used to train their voting classifier. Their classifier incorporates textual, shape, color, and texture features.

A slightly more automatic process, termed "OPTIMOL," is used by Li *et al.* [10]. Exploiting the fact that the first few results from search engines tend to be very good, they use the first 5-10 images returned by Google Image Search to train their classifier, built upon a Hierarchical Dirichlet Process [11]. OPTIMOL considers images sequentially, classifying them as either background or relevant. If the image is classified as relevant, the classifier uses incremental learning to refine its model. As the classifier accepts more images, these images allow the classifier to obtain a richer description of the category.

Most recently, the fully automatic "Harvesting Image Databases from the Web" by Schroff *et al.* [12] uses text information to re-rank images retrieved from text-based search. The top-ranked images form the training set of a support vector machine, which relies on visual information to re-rank images once again.

Finally, the Tiny Images [13] project aims to crawl a massive number of images from the internet. They have collected 80 million images to date, for about 75,000 keywords. The goal of this dataset is to collect an extensive collection of images, rather than to obtain high accuracy for each keyword.

2 Approach

Our general approach is motivated primarily by accuracy and scalability. We aim to deal with image categories with very many candidate images, exhibiting intraclass variation. High degrees of intraclass variability normally suggest that a large and accurate training set is necessary; our approach also aims to minimize the time required of the user while still capturing large amounts of diversity in the dataset.

2.1 Crawling

We rely on image search engines to obtain our noisy image set, leveraging the tremendous number of images available on the web. As noted by Schroff *et al.*, most image search engines restrict the number of images returned. To overcome this restriction, we generate multiple related queries using a standard lexical database [14]. We also translate our queries into several languages, accessing the regional website of our image search engines (e.g., *http://images.google.cn*).

2.2 Learning

We utilize a discriminative learning scheme with active, online learning. Our learning procedure begins as the user labels several randomly chosen images

(on the order of several dozen). Our classifier learns on these images, and then examines the set of unlabeled images to select an additional set of images to be labeled. Once these images are labeled, the classifier trains on the newly labeled images, and again selects more images to be labeled. This process proceeds iteratively until sufficient classification accuracy is obtained, as determined by the user.

At the outset, it may seem curious to choose a supervised approach in light of the recent research emphasizing nontraditional supervision (as in Berg *et al.*) or fully automatic operation (as in Li *et al.* and Schroff *et al.*). As shown in the experiments of this work and its predecessors, even state-of-the-art techniques do not always achieve the level of performance necessary for large-scale, high-precision datasets. As such, our aim is to demonstrate performance which is superior to recent research with minimal levels of supervision, while also allowing the user to apply greater levels of supervision in an opportunistic fashion.

Confidence-Weighted Boosting. The basis of our discriminative learning scheme is confidence-weighted boosting [15]. Confidence weighted boosting is similar to AdaBoost [16], but differs in that weak learners yield a real-valued vote instead of a binary vote. In our case, we take as our weak learner decision stumps. Each potential decision stump partitions the training data into two disjoint sets. For a given set $i \in \{1, 2\}$, we let W_+^i be the sum of the weight of positive instances in the set, and likewise for W_-^i. In each round of boosting, we select the decision stump which minimizes the quantity

$$Z = \sum_{i \in \{1,2\}} \sqrt{W_+^i W_-^i} \tag{1}$$

Using these quantities, we can also determine the manner in which the stump will vote (as its decision is no longer a binary +1 or -1). Given that a particular instance falls into partition i, the classifier's vote for that instance is given by

$$c_i = \frac{1}{2} \ln \left(\frac{W_+^i}{W_-^i} \right) \tag{2}$$

Once a weak learner is selected, the weights of each training instance are updated as in AdaBoost. Our choice of confidence-weighted boosting was motivated by its excellent speed and accuracy.

Active Learning. Because the number of unlabeled images greatly outnumbers the number of labeled ones, it is natural to try to exploit the unlabeled data in some way. Active learning is one technique to do so, as it allows the machine learning engine to select a subset of the unlabeled data to be labeled. Our approach is to apply the learned classifier to the set of unlabeled data, and select the subset of images for which the predicted class is least certain (i.e., those for which the sum of the votes of our classifiers is closest to zero). The active learning method has been applied in vision to classify videos [17,18]; our boosting-based approach is most similar to that applied by Tur *et al.* [19] in a speech-processing application.

Active learning is particularly well-suited to the sort of data we expect to see from an internet crawler, as there will be many images which are highly similar (if not near-duplicates). It serves little good to label images which are very likely positive given the training data; active learning allows us to focus the user's attention on the examples which add richness and diversity to the dataset.

Online Learning. Each time we obtain a new set of labeled images, a naive approach might be to take the set of all images labeled thus far as our training set, and run our algorithm as usual. However, this discards everything which has been learned from previous stages of our active learning process. Several sophisticated online learning schemes have been proposed for boosting ([20,21]), but we consider two simple, heuristic schemes.

In the first scheme, we simply set our set of weak classifiers to be those obtained on the smaller set of data. We weight each new training instance as though it had been present during the previous stage of learning, though the weighted votes of our classifiers do not reflect the presence of these datapoints. We then apply the AdaBoost algorithm as usual for several rounds of boosting, in order to learn modalities not present in the smaller dataset.

In a slightly more sophisticated scheme, learning proceeds in two stages. In the first stage, we restrict the universe of weak classifiers to those which were obtained on the smaller set of labeled images. Boosting proceeds as usual, and the real-valued votes of these classifiers is recomputed to reflect the new dataset at each round of boosting. However, learning operates much more quickly at this stage than it normally would, as far fewer weak classifiers need be considered. Once a number of weak classifiers have been obtained using this method, we apply the AdaBoost as usual. As our experiments show, this method gives superior performance to the naive scheme at only marginally greater computational cost.

3 System Overview: Walkthrough of the Ape Category

In this section, we provide a walkthrough of the ape image category. Our crawling technique yields 21526 images, of which 5292 actually contain an image of an ape. Throughout this paper, we consider abstract images (e.g., drawings of apes, pictures of toys in the form of an ape) to be background images.

First, a descriptor vector is computed. We begin by extracting descriptors of several types: SIFT codeword histograms, filterbank response histograms, color histograms, downsampled pixel values, and search engine rank. The complete descriptor vector for an image is the concatenation of the vector obtained through the following methods:

- **SIFT codeword histograms:** A codebook of SIFT descriptor vectors [22] is formed by first extracting SIFT descriptors from a random subset of the crawled images (on the order of 1000 images). We run Fast K-means [23] to obtain 300 128-dimensional vectors, which form our codebook. We then again run the SIFT algorithm on each image in the crawled dataset, and match each keypoint descriptor into the codebook by choosing the codeword which

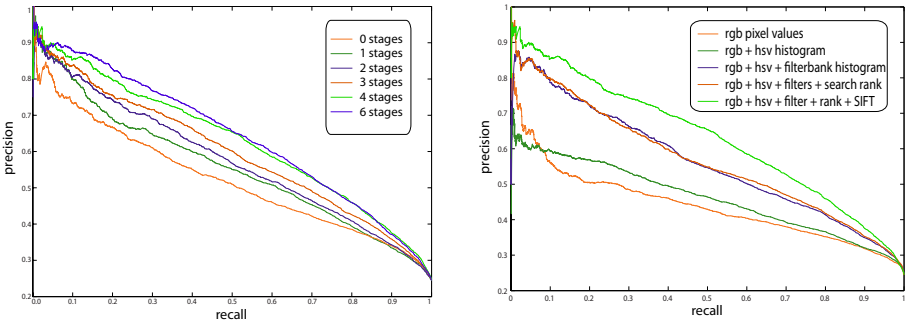
minimizes Euclidean distance. Finally, we compute a normalized histogram of codeword expression for each image.

- **Filterbank response histograms:** To obtain filterbank response histograms, we first convolve each image with each of 48 kernels, taken from the LM-48 filterbanks [24]. For each convolution response, we form an 11-bin histogram of responses.
- **Color histograms:** The color histogram descriptor for an image is simply a histogram of color expression in HSV-space.
- **Downsampled Pixel values:** Inspired by Torralba *et al.* [13], this descriptor is formed by simply applying histogram equalization, downsampling an image to 16x16 pixels, and giving the intensity of each pixel in RGB-space.
- **Image rank:** Finally, we preserve the rank order of each image in the search engines from which is was crawled. If an image was found on multiple search engines, this descriptor yields the minimum rank obtained. The motivation for this feature is that we expect our search engines to be effective at text processing, but to neglect image processing. We can incorporate the result of their text processing into our feature set without adding computational complexity to our pipeline.

Once these descriptors have been computed, learning begins. Figure 1 shows the process from the users' point of view: the initial set of randomly chosen images, the set of images chosen for active learning, and the classifier's top-ranked images after one stage of active learning. Note the types of images chosen for active learning: they can reflect some elements common in ape images (ape-like textures, natural scenery), but the positive images reflect elements less common for ape images, such as unusual viewpoints or scales.

Figure 2 shows how the performance of our classifier improves as stages of active learning progress, and shows the importance of each feature in our descriptor vector.

Fig. 1. Labeling process from user's point of view. **(left)** is a subset of 50 randomly chosen initial images. **(center)** shows some of the images selected for labeling after the first stage of learning. **(right)** shows highly-ranked images after 1 round of active learning.



(a) (b)

Fig. 2. Classification performance **(a)** by stages of active learning and **(b)** by feature set. This figure is best viewed in color.

4 Experiments and Results

In this section, we provide results of our learning algorithms in comparison to simpler alternatives, and also compare our whole-system performance to that of Berg *et al.*, Li *et al.*, and Schroff *et al.*

4.1 Performance of Learning Algorithms

Figure 3 provides convincing evidence that active learning dramatically reduces the number of labeled examples necessary to obtain high-precision classification. In each case we begin with 100 randomly chosen images; as active learning selects more images in a incremental fashion, the disparity between its performance benefit over passive learning increases. In particular, active learning with 250 labeled images outperforms passive learning with 400 labeled images.

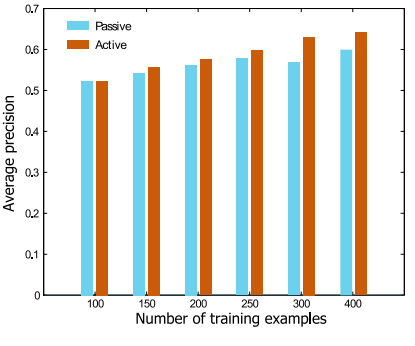


Fig. 3. Performance comparison of active learning and passive learning, under various numbers of training examples. In the case of active learning, we begin with 100 randomly chosen images, with the remaining labeled images chosen actively in increments of 50 images.

Table 1. Time and Classification Performance of Online Learning System. We evaluate on the ape dataset with 400 training examples, of which 300 were chosen actively. At each stage the number of rounds of boosting is proportional to the number of training images. For the online-naive approach, all weak learners from the previous stage are retained. For our online-relearning scheme, 50% of the weak learners at any stage are drawn from the weak learners of the previous stage, and the remainder are computed normally.

Algorithm	time	Area under precision-recall curve
Online-naive	91 s	.5743
Online-relearning	144 s	.6281
Batch relearning	221 s	.6415

Moreover, our online-learning approach yields improved time-complexity without hurting performance (table 1).

4.2 Comparison with OPTIMOL

In this section we compare our performance to that of Li *et al.*, using code provided by the authors. The training set for our algorithm consists of 200 images (100 images are chosen randomly; the remainder are chosen through two stages of active learning). OPTIMOL's supervision scheme is somewhat different, taking only a positive seed set. We thus feed OPTIMOL the positive images from our training set. The result of this comparison is presented in Figure 4.

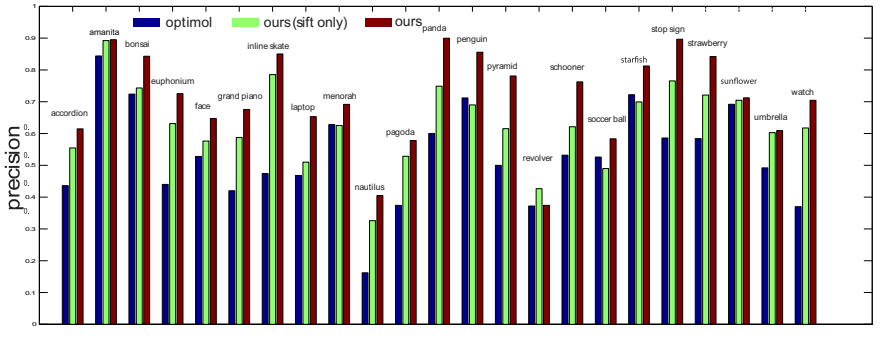


Fig. 4. Performance Comparison of OPTIMOL and our algorithm. We evaluate our algorithm using the complete feature set, and using only the SIFT codeword histograms. OPTIMOL is trained on the positive images from our training set. Our superior performance shows the effectiveness both of our learning mechanism and our feature set.

As a result of the sequential nature of OPTIMOLs classifier, there is no natural way to generate precision-recall curves. Instead, OPTIMOL returns a set of putatively positive images, the size of which cannot be specified *a priori*. We present the precision of this set of returned by OPTIMOL. To establish an

equal comparison, we obtain an equal-sized set of top-ranked images from our algorithm, and present the precision of this dataset. In order to evaluate the extent to which our performance difference is dependent on our feature set, we also present performance of our algorithm when our only feature is SIFT codeword histograms. As can be seen, in 22 of the 23 categories, the performance of our algorithm is superior to that of OPTIMOL. Moreover, in 20 of the 23 categories, our algorithm operating only on the SIFT histograms is superior, suggesting that our learning methodology is better-suited to this application than is that of OPTIMOL.

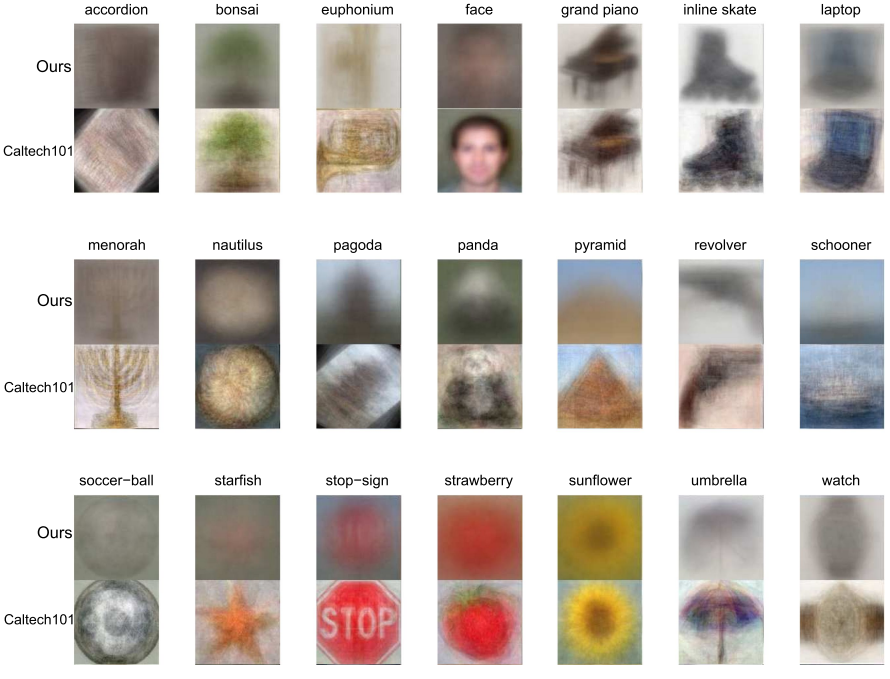


Fig. 5. Average image of our collected dataset, in comparison to that of Caltech-101. For each pair of rows, our average image is the top-most. False positives are not included in our average image. Our approach yields a diverse dataset, as active learning allows rapid learning of the most challenging cases.

We also present the relative runtimes of our systems in Figure 6. It is clear that our time performance is far superior, despite a greater feature set. Each round of boosting takes 5 ms per image, compared to 1.7 s for OPTIMOL. The significant speed advantage of our learning algorithm allows us to explore more descriptive feature sets; indeed, AdaBoost is "free" in comparison to feature extraction in terms of computational time.

To illustrate the diversity of the images we collect, Fig. 5 shows the average image comparison between the Caltech101 dataset [1] and our newly collected images. The images we collect show greater interclass variation. In short, our

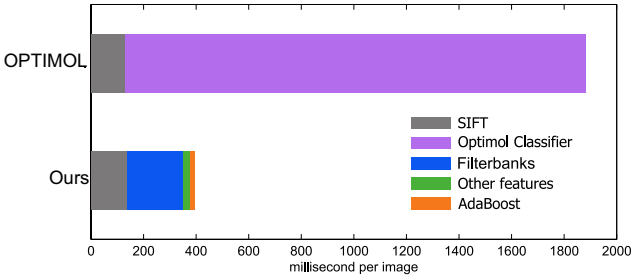


Fig. 6. Processing time of our algorithm and of OPTIMOL, per image, averaged over all categories. Because our learning algorithm is drastically faster than that of OPTIMOL, we are able to consider a richer feature set while still maintaining a high degree of scalability. This figure is best viewed in color.

approach has superior precision and superior time performance in comparison to OPTIMOL. As such, our approach shows much greater promise as a system scalable to very large datasets.

4.3 Evaluation on Animals on the Web Dataset

Two other major approaches, Berg *et al.* and of Schroff *et al.*, present results on the Animals on the Web (AoW) dataset; we consider our performance on this dataset.

Figure 7 presents the classification accuracy on the Berg dataset. Our results reflect only test data, given 150 labeled images. The first 50 images were selected randomly; the remaining 100 were selected using active learning in two stages. We present precision-recall curves for a number of these categories in Figure 8.

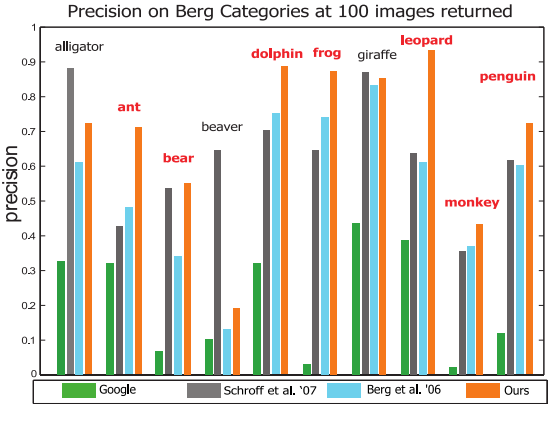
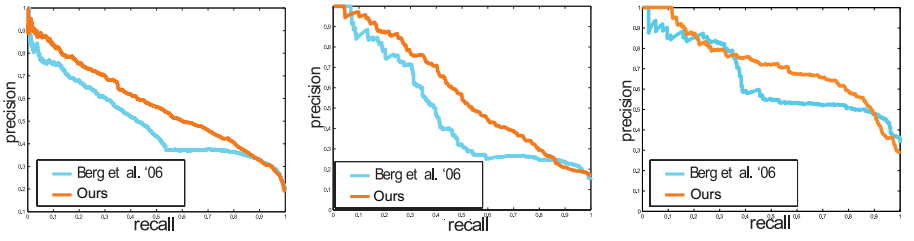


Fig. 7. Our precision on the Animals on the web dataset, in comparison to Schroff *et al.* and Berg *et al.* In seven of ten categories, our precision is superior; we mark these categories with red labels.



(a) monkey (b) frog (c) giraffe

Fig. 8. Precision-recall graphs for the categories of monkey, frog, and giraffe. Berg *et al.*'s performance is colored cyan; our results are in orange.

It is worth dwelling for a moment on the experimental conditions of Berg *et al.* and of Schroff *et al.*. Like Schroff, we compare our performance to that of the AoW test data, as results presented under "final dataset" include those directly labeled by the user. We believe that the amount of supervision is comparable between Berg *et al.* and this work; our user is asked to examine 150 images and will (assuming 25% precision of the crawled data) click on approximately 40 of them, as the user is only required to click on positive images. Similarly, Berg *et al.* present 10 topics, each of 30 images. The user in Berg *et al.* has the option of only clicking on 10 images. However, it is clear from their Figure 2 that much higher levels of high-precision recall are obtained when the additional step of swapping images between topics is performed. The Berg *et al.* performance we compare against reflect this additional supervision. The number of images examined by the user, as well as the number of images actually clicked, is comparable to that of Berg *et al.* [9].

It is more difficult to assess matters with respect to Schroff *et al.*. Though they indeed are fully automatic, they also avail themselves of a much larger, automatically harvested training set. It is likely that this aides them greatly in categories such as beaver, where the precision of the underlying dataset is very low. Though our training sets are smaller than those of Schroff *et al.*, minimal user supervision, targeted at the most informative examples using active learning, can provide performance which is comparable to the much larger training sets of their approach.

On seven of ten categories, our algorithm gives superior precision to both Schroff *et al.* and Berg *et al.*. In all ten categories, our performance is superior to that of Berg *et al.*. The precision-recall graphs of Figure 8 are also informative. Consider in particular the monkey dataset, as it is much larger than the others. The superior precision of our approach across the recall curve is testament to its scalability — users of our algorithm can expect to achieve very competitive performance regardless of the level of recall they desire.

4.4 Mammal Dataset

In order to demonstrate the scalability of our dataset, we aim to replicate the dataset of Fink and Ullman [5], which includes several esoteric mammal species. The Fink dataset consists of 400 mammalian image categories, with the candidate images obtained using Google Image Search. For each animal, the images are divided into five categories: irrelevant images, images which are not color photographs (which includes abstract images), color images which include a cropped version of the animal, color images with the full animal in a non-standard pose or view, and finally color images which include the animal in a standard pose.

We have replicated this dataset using the procedure presented in this paper, and made it available at `http://vision.cs.princeton.edu/easymammal.htm`. We do not separate our images into tiers as does Fink *et al.*; we consider his final three categories, plus black and white images, to be positive. Similarly, we do not provide the detailed annotation of his dataset. The labeling we employ consists of 100 randomly chosen images, followed by two stages of active learning with 50 images in each stage. Using an interface which requires the user to click on the positive images, it takes a user approximately 3 minutes to accomplish all three stages.

Table 2. For several mammal categories, we present the number of images present in the Fink *et al.* dataset, the number of images we obtain through crawling, and the precision we obtain at 200 images returned. For five of the ten categories, we obtain more images than Fink *et al.* within the first 200 images returned (precision bolded for these categories). We are very successful in the categories in which Fink obtains high recall ($n > 200$), with precision of 0.97 in four of five such categories.

Category	# of images in Fink	# of images crawled	Precision at 200 images returned
Aardvark	48	23152	**.885**
Camel	160	15951	**.95**
Fox	39	15842	**.925**
Chipmunk	301	14571	.975
Monkey	94	36765	**.94**
Hyena	293	19915	.97
Spotted Hyena	250	9170	.975
Brown Bear	213	25762	.91
Lion	60	27032	**.99**
Bengal Tiger	280	8981	.985

Here we highlight several categories. Table 2 gives the number of candidate images we obtained in comparison to Fink. Figure 9 shows the average image of the top 200 images returned. The superior number of crawled results (in comparison to the 1,000 maximum images returned by a single query to Google Image Search) is due to our use of query expansion (including latin scientific names, in most cases), and foreign language translation. Further, despite the limited

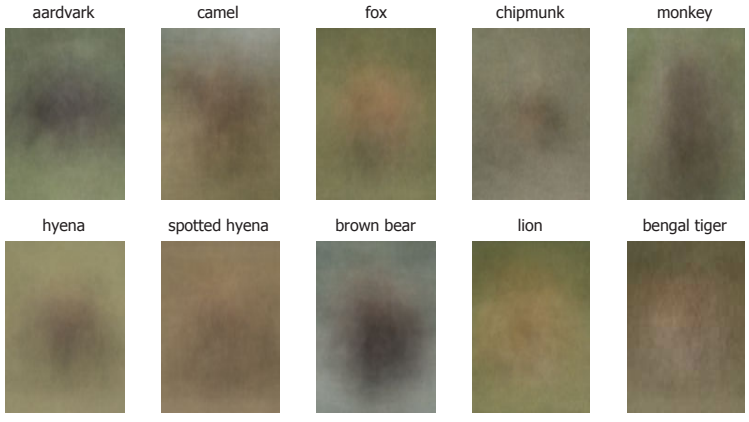


Fig. 9. Average image of the top 200 results for the ten animal categories in Table 2. False positives are not included in.

supervision we require, the precision we provide is quite high. A very rapid post-processing step could remove the false positives from this data, resulting in a large, diverse dataset with minimal effort on the part of the user.

5 Discussion

We have presented and evaluated a scalable and accurate dataset construction technique. Our diverse feature set and accurate machine learning technique allow for precision which is superior to the state-of-the-art. Moreover, the use of active learning serves to minimize the need for supervision, while allowing the user to opportunistically apply labels where necessary to improve the precision of the dataset. Finally, our use of features which are easy to compute efficiently and online learning allows for superior computational complexity. For the future, we intend to continue developing both the learning technique and the feature representations to improve our classification accuracy. It is also worthwhile to push for faster algorithms that can achieve real-time learning while users annotate.

References

1. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories (2004)
2. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset (2007)
3. Everingham, M., Zisserman, A., Williams, C.K.I., Van Gool, L.: The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results, `http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf`
4. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: A database and web-based tool for image annotation. Int. J. Comput. Vision 77, 157–173 (2008)

5. Fink, M., Ullman, S.: From aardvark to zorro: A benchmark for mammal image classification. Int. J. Comput. Vision 77, 143–156 (2008)
6. Yao, B., Yang, X., Zhu, S.C.: Introduction to a large-scale general purpose ground truth database: Methodology, annotation tool and benchmarks, pp. 169–183 (2007)
7. Fergus, R., Perona, P., Zisserman, A.: A visual category filter for google images. In: Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic, pp. 242–256 (2004)
8. Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A.: Learning object categories from google's image search. In: Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005, 17-21 October 2005, vol. 2, pp. 1816–1823 (2005)
9. Berg, T.L., Forsyth, D.A.: Animals on the web. Computer Vision and Pattern Recognition, 1463–1470 (2006)
10. Li, J., Wang, G., Fei-Fei, L.: Optimol: automatic object picture collection via incremental model learning. Computer Vision and Pattern Recognition (2006)
11. Teh, Y.W., Jordan, M.I., Beal, M.J., Blei, D.M.: Hierarchical dirichlet processes. Journal of the American Statistical Association (2006)
12. Schroff, F., Criminisi, A., Zisserman, A.: Harvesting image databases from the web (2007)
13. Torralba, A., Fergus, R., Freeman, W.T.: Tiny images. Technical Report MIT-CSAIL-TR-2007-024, Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology (2007)
14. Princeton Cognitive Science Laboratory: Wordnet, `http://wordnet.princeton.edu`
15. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Mach. Learn. 37, 297–336 (1999)
16. Schapire, R.: The boosting approach to machine learning: An overview. In: Hansen, M., Holmes, C., Mallick, B., Yu, B. (eds.) Nonlinear Estimation and Classification. Springer, Heidelberg (2003)
17. Yan, R., Yang, J., Hauptmann, A.: Automatically labeling video data using multi-class active learning. In: Eighth IEEE International Conference on Computer Vision. ICCV 2003, vol. 01, p. 516 (2003)
18. Abramson, Y., Freund, Y.: Semi-automatic visual learning (seville): a tutorial on active learning for visual object recognition. In: Computer Vision and Pattern Recognition (2005)
19. Hakkani-Tür, D., Riccardi, G., Tur, G.: An active approach to spoken language processing. ACM Trans. Speech Lang. Process 3, 1–31 (2006)
20. Oza, N.: Online bagging and boosting. In: IEEE International Conference on Systems, Man and Cybernetics, vol. 3, pp. 2340–2345 (2005)
21. Grabner, H., Bischof, H.: On-line boosting and vision. In: CVPR 2006: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, pp. 260–267. IEEE Computer Society Press, Los Alamitos (2006)
22. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (2004)
23. Elkan, C.: Using the triangle inequality to accelerate k-means. In: Proceedings of the Twentieth International Conference on Machine Learning, pp. 147–153 (2003)
24. Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons (2001)

GeoS: Geodesic Image Segmentation

Antonio Criminisi, Toby Sharp, and Andrew Blake

Microsoft Research, Cambridge, UK

Abstract. This paper presents GeoS, a new algorithm for the efficient segmentation of n-dimensional image and video data.

The segmentation problem is cast as approximate energy minimization in a conditional random field. A new, parallel filtering operator built upon efficient geodesic distance computation is used to propose a set of spatially smooth, contrast-sensitive segmentation hypotheses. An economical search algorithm finds the solution with minimum energy within a sensible and highly restricted subset of all possible labellings.

Advantages include: i) computational efficiency with high segmentation accuracy; ii) the ability to estimate an approximation to the posterior over segmentations; iii) the ability to handle generally complex energy models. Comparison with max-flow indicates up to 60 times greater computational efficiency as well as greater memory efficiency.

GeoS is validated quantitatively and qualitatively by thorough comparative experiments on existing and novel ground-truth data. Numerous results on interactive *and* automatic segmentation of photographs, video and volumetric medical image data are presented.

1 Introduction

The problem of image and video segmentation has received tremendous attention throughout the history of computer vision with excellent recent results being achieved both interactively [1,2] and automatically [3]. However, state of the art techniques are not fast enough for real-time processing of high resolution images (e.g. $> 2Mpix$). This paper describes a new, efficient algorithm for the accurate segmentation of n-dimensional, high-resolution images and videos.

Like many vision tasks, the segmentation problem is usually cast as energy minimization in a Conditional Random Field (CRF) [1,2,4,6,7]. This encourages spatial-smoothness and contrast-sensitivity of the final segmentation. The same framework is employed here; but in contrast to graph cut-based approaches here the segmentation is obtained as the labeling corresponding to the energy minimum (MAP solution) found within a restricted, sensible subset of all possible segmentations. Such a solution will be shown to be smooth and edge aligned. The segmentation posterior over the selected subspace can also be estimated, thus enabling principled uncertainty analysis (see also [5]). Quantitative comparisons with ground truth will demonstrate segmentation accuracy equal or superior

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 99–112, 2008.

to that of the global minimum as found by min-cut/max-flow[1]. Restricting the search space to a small, sensible one accounts for the computational efficiency.

Similar to the work of Bai et al. in [9], we also use geodesic transforms to encourage spatial regularization and contrast-sensitivity. However, GeoS differs from [9] in a number of ways: i) The technique in [9] assumes given user strokes and imposes an implicit connectivity prior which forces each region to be connected to one such stroke[2]. In contrast, our geodesic filter acts on the energy unaries (not the user strokes). This allows GeoS to generate segmentations with no topological restrictions. ii) GeoS is not specific to *interactive* segmentation and can be applied to *automatic* segmentation as well as other tasks such as denoising, stereo and panoramic stitching. iii) GeoS presents a clear energy to be minimized. This allows quantitative comparisons with other energy-based approaches such as GrabCut [2]. Finally, iv) Despite the complexity of both algorithms being optimally linear in the number of pixels, GeoS, thanks to its contiguous memory access and parallelism is much faster than [9] in practice.

Efficient segmentation via energy minimization has also been the focus of the dual-primal technique in [10] and the logarithmic α-expansion scheme in [11]. In spatio-temporal MRFs efficiency may be gained by either reusing the graph flow or the search trees [12,13]. Instead, the efficiency of GeoS stems from its optimized memory access and its ability to exploit the power of modern multi-core architectures. In contrast, graph-cut does not lend itself to easy parallelization.

Finally, unlike graph-cut, our approximate minimization algorithm is not restricted to a specific family of energies. This enables us to experiment with more sophisticated models, like those containing *global* constraints.

2 Background on Distance Transforms

This section presents background on geodesic distances and related algorithms.

Unsigned geodesic distance. Given an image I defined on a 2D domain Ψ, a binary mask M (with $M(\mathbf{x}) \in \{0,1\} \forall \mathbf{x}$) and an "object" region Ω with $\mathbf{x} \in \Omega \iff M(\mathbf{x}) = 0$, the unsigned geodesic distance of each pixel $\mathbf{x}$ from Ω is defined as:

$$D(\mathbf{x}; M, \nabla I) = \min_{\{\mathbf{x}' | M(\mathbf{x}')=0\}} d(\mathbf{x}, \mathbf{x}'), \quad \text{with} \tag{1}$$

$$d(\mathbf{a}, \mathbf{b}) = \min_{\boldsymbol{\Gamma} \in \mathcal{P}_{\mathbf{a},\mathbf{b}}} \int_0^1 \sqrt{||\boldsymbol{\Gamma}'(s)||^2 + \gamma^2 \left(\nabla I \cdot \mathbf{u}\right)^2} \; ds \tag{2}$$

with $\mathcal{P}_{\mathbf{a},\mathbf{b}}$ the set of all paths between the points $\mathbf{a}$ and $\mathbf{b}$; and $\boldsymbol{\Gamma}(s) : \Re \rightarrow \Re^2$ indicating one such path, parametrized by $s \in [0,1]$. The spatial derivative $\boldsymbol{\Gamma}'(s)$

[1] The fact that a local energy minimum may be more accurate than the global one should not come as a surprise. In fact [8] have discussed the limitations of the widely used unary + pairwise energies and the need for more realistic models.

[2] E.g. in [9] segmenting the image of a chess board into its black and white squares would require $8 \times 8 = 64$ user strokes.

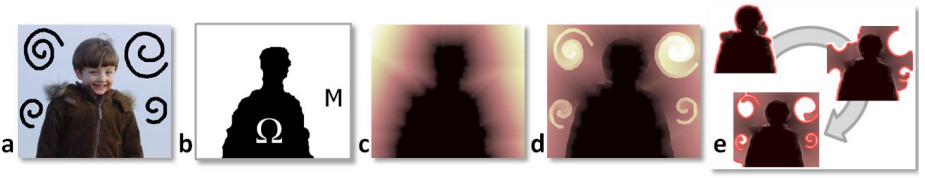


Fig. 1. $\mathcal{O}(N)$ **geodesic distance transform algorithms. (a)** Original image, I; **(b)** Input mask M with "object" Ω. **(c)** Distance $D(\mathbf{x}; M, \nabla I)$ from Ω (with $\gamma = 0$ in (1)); **(d)** *Geodesic* distance from object ($\gamma > 0$) computed with the raster scan algorithm in [18] (two complete raster-scan passes suffice). Note the large jump in the distance D in correspondence with strong edges. **(e)** Different stages of front propagation of the algorithm in [19], eventually leading to a geodesic distance similar to the one in (d).

is $\boldsymbol{\Gamma}'(s) = \partial \boldsymbol{\Gamma}(s)/\partial s$. Also, the unit vector $\mathbf{u} = \boldsymbol{\Gamma}'(s)/||\boldsymbol{\Gamma}'(s)||$ is tangent to the direction of the path. The factor γ weighs the contribution of the image gradient versus the spatial distances. Equation (1) generalizes the conventional Euclidean distance; in fact, D reduces to the Euclidean path length for $\gamma = 0$.

Distance transform algorithms. Excellent surveys of techniques for computing *non-geodesic* distance transforms may be found in [14,15]. There, two main kinds of algorithms are described: *raster-scan* and *wave-front propagation*. Raster-scan algorithms are based on kernel operations applied sequentially over the image in multiple passes [16]. Instead, wave-front algorithms such as Fast Marching Methods (FMM) [17] are based on the iterative propagation of a pixel front with velocity F.

Geodesic versions of both kinds of algorithms may be found in [18] and [19], respectively. An illustration is shown in fig. 1. Both the Toivanen and Yatziv algorithms produce approximations to the actual distance and both have optimal complexity $\mathcal{O}(N)$ (with N the number of pixels). However, this does not mean that they are equally fast in practice. In fact, FMM requires accessing image locations far from each other in memory. Thus, the limited memory bandwidth of modern computers limits the speed of execution of such algorithms much more than their modest computational burden. In contrast, Toivanen's technique (employed here) accesses the image memory in *contiguous* blocks, thus minimizing such delays. This yields speed up factors of at least one order of magnitude compared to [19]. Algorithmic details are presented in the Appendix.

3 Geodesic, Symmetric Morphology

This section introduces a new filtering operator which constitutes the basis of our segmentation process. The filter builds upon efficient distance transforms.

Geodesic morphology. The two most basic morphological operations – erosion and dilation – are usually defined in terms of binary structured elements acting on binary images. However, it is possible to redefine those operations as functions of real-valued image distances, as follows. Equation (1) leads to the following definition of the *signed* geodesic distance from the object *boundary*:

$$D_s(\mathbf{x}; M, \nabla I) = D(\mathbf{x}; M, \nabla I) - D(\mathbf{x}; \overline{M}, \nabla I), \tag{3}$$

with $\overline{M} = 1 - M$. It follows that dilation and erosion may be obtained as

$$M_d(\mathbf{x}) = [D_s(\mathbf{x}; M, \nabla I) > \theta_d], \quad M_e(\mathbf{x}) = [D_s(\mathbf{x}; M, \nabla I) > -\theta_e] \tag{4}$$

with $\theta > 0$ the diameter of the disk-shaped structured element. The indicator function [.] returns 1 if the argument is true and 0 otherwise. More useful, *idempotent* filters (an operator f is idempotent iff. $f(f(x)) = f(x)$) such as closing and opening are achieved as:

$$M_c(\mathbf{x}) = [D(\mathbf{x}; \overline{M_d}, \nabla I) > -\theta_e], \quad M_o(\mathbf{x}) = [D(\mathbf{x}; M_e, \nabla I) > \theta_d] \tag{5}$$

respectively. Redefining known morphological operators in terms of real-valued distances allows us to: i) implement those operators very efficiently, and ii) introduce contrast sensitivity effortlessly, by means of geodesic processing. Next, a further modification to conventional morphology is introduced.

Adding symmetry. Closing and opening are asymmetrical operations in the sense that the final result depends on the order in which the two component operations are applied to the input mask (see fig. 2g,h). However, in image filtering one would just wish to define the dimension of the regions to be removed (e.g. noise speckles) and apply the filter without worrying about the sequentiality of operations within the filter. Here we solve this problem by defining the following new, symmetrical filter:

$$M_s(\mathbf{x}; M, I) = [D_s^s(\mathbf{x}; M, \nabla I) > 0] \tag{6}$$

where the symmetric, signed distance D_s^s is defined as:

$$D_s^s(\mathbf{x}; M, \nabla I) = D(\mathbf{x}; M_e, \nabla I) - D(\mathbf{x}; \overline{M_d}, \nabla I) + \theta_d - \theta_e, \tag{7}$$

with M_e and $\overline{M_d}$ defined earlier. The additional term $\theta_d - \theta_e$ enforces the useful idempotence property; *i.e.* it keeps unaltered the remaining signal structure. Formulating morphological operations in terms of real-valued distances allows us to perform symmetrical mixing of closing and opening via (7). The only two geometric parameters θ_d, θ_e are very intuitive as they correspond to the maximum size of the foreground and background noise speckles to be removed.

In summary, the operator (6) generalizes existing morphological operations by adding symmetry and edge-awareness. In fact, setting $\gamma = 0$ and then $\theta_d = 0$ ($\theta_e = 0$) reproduces conventional closing (opening). Figure 2 illustrates the filtering process for 1D and 2D toy examples. Isolated peaks and valleys are simultaneously removed while maintaining unaltered the remaining signal. Equipped with this new tool we can now focus on the segmentation problem.

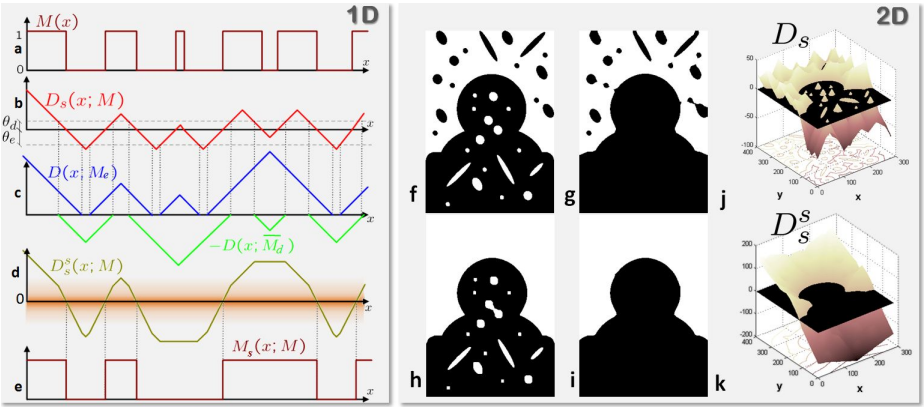


Fig. 2. Symmetric filtering in 1D and 2D. (a) Input, binary 1D signal M. **(b)** The initial signed distance D_s. **(c)** The two further unsigned distances for selected values of θ_d, θ_e. **(d)** The final signed distance D_s^s. **(e)** The filtered mask $M_s(x;M)$. Some of the peaks and valleys of $M(x)$ have been removed while maintaining the integrity of the remaining signal. For simplicity of explanation here no image gradient is used. Now let's look at a 2D example. **(f)** Original 2D mask M, **(g)** mask after closing, **(h)** after opening, **(i)** resulting mask M_s after our symmetric filtering. **(j)** The distance $D_s(\mathbf{x})$ for the input 2D mask in (f). **(k)** The final distance $D_s^s(\mathbf{x})$ for $\theta^d = 10$ and $\theta^e = 11$. The intersection of $D_s^s(\mathbf{x})$ with the xy plane through 0 results in the filtered mask M_s shown in (i). The parameters θ_d and θ_e are fixed in all (f,...,i).

4 Segmentation Via Restricted Energy Minimization

The binary segmentation problem addressed here is cast as minimizing an energy of type

$$E(\mathbf{z}, \boldsymbol{\alpha}) = U(\mathbf{z}, \boldsymbol{\alpha}) + \lambda V(\mathbf{z}, \boldsymbol{\alpha}) \tag{8}$$

with $\mathbf{z}$ the image data and $\boldsymbol{\alpha}$ the per-pixel labeling, with $\alpha_n \in \{\texttt{Fg}, \texttt{Bg}\}$. The subscript n indexes the pixels and `Fg` (`Bg`) indicates foreground (background). The unary potential U is defined as the sum of pixel-wise likelihoods of the form $U(\mathbf{z}, \boldsymbol{\alpha}) = -\sum_n \log p(z_n|\alpha_n)$; and the data-dependent pairwise term is $V(\mathbf{z}, \boldsymbol{\alpha}) = -\sum_{m,n\in\mathcal{N}} [\alpha_n \neq \alpha_m] \exp(-|z_n - z_m|/\eta)$. Here we use 8-neighborhood cliques $\mathcal{N}$. Flux may also be incorporated in (8) as a further unary term.

Sub-modular energies of the form (8) can be minimized exactly by min-cut. However, in image segmentation, finding the global minimum of such energy makes sense only provided that the energy model correctly captures the statistics of natural images. Recent work has shown that this is often *not* the case [8]. It has been observed that often local energy minima correspond to segmentations which are more accurate (compared to ground truth) than that yielded by the global minimum. Thus, a technique that can find good local minima efficiently becomes valuable. This section describes such an approximate and efficient technique. Later we will also show how such algorithm can be applied to energy models of a more general nature than the one in (8).

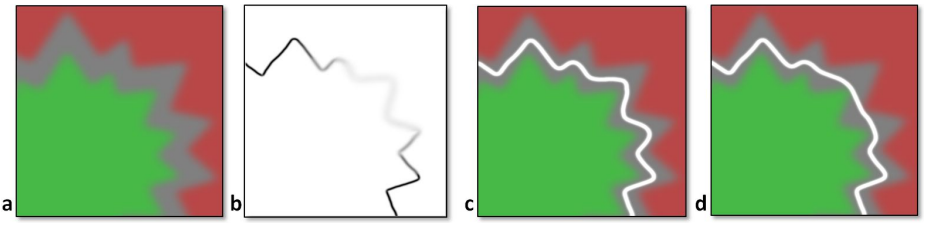


Fig. 3. Filter behaviour in the presence of weak unaries. (a) Input unaries (green for `Fg` and red for `Bg`), with large uncertain areas (in grey). **(b)** Magnitude of gradient of input image. **(c)** Computed segmentation boundary (white curve) for a *small* value of $\theta_d = \theta_e$. **(d)** As in (c) but for *large* θ. Larger values of θ yield smoother segmentation boundaries in the presence of weak edges and/or weak unaries. In contrast, strong gradients "lock" the segmentation in place.

Key to our algorithm is the minimization of the energy in (8) by efficient search of the solution $\boldsymbol{\alpha}^*$ over a restricted, parametrized 2D manifold of all possible segmentations. Let us define $\boldsymbol{\theta} = (\theta_d, \theta_e) \in \mathcal{S}$, with $\mathcal{S} \subset \mathbb{R}^2$. As described earlier, given a value of $\boldsymbol{\theta}$ the geodesic operator (6) has the property of removing isolated regions (with dimensions $< \boldsymbol{\theta}$) from foreground and background in binary images. Therefore, if we can adapt our filter to work on real-valued unaries, then for different values of $\boldsymbol{\theta}$ different levels of spatial smoothness would be obtained and thus different energy values. The segmentation we are after is

$$\boldsymbol{\alpha}^* = \boldsymbol{\alpha}(\boldsymbol{\theta}^*), \quad \text{with} \quad \boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \mathcal{S}} E(\mathbf{z}, \boldsymbol{\alpha}(\boldsymbol{\theta})).$$

Next we focus on the details of the GeoS algorithm.

Segmentation proposals. In a binary segmentation problem, given the *real*-valued log likelihood ratio: $L(\mathbf{x}) = \log p(z_{n(\mathbf{x})}|\alpha_{n(\mathbf{x})} = \texttt{Fg}) - \log p(z_{n(\mathbf{x})}|\alpha_{n(\mathbf{x})} = \texttt{Bg})$ we redefine the mask $M(\mathbf{x}) \in [0, 1]$ as a log-odds map $M(\mathbf{x}) = \sigma(L(\mathbf{x}))$ with $\sigma(.)$ the sigmoid transformation $\sigma(L) = 1/(1 + \exp(-L/\mu))$ [3]. The distance (1) then becomes:

$$D(\mathbf{x}; M, \nabla I) = \min_{\mathbf{x}' \in \Psi} (d(\mathbf{x}, \mathbf{x}') + \nu M(\mathbf{x}')) \tag{9}$$

with $d(.)$ as in (2). ν (trained discriminatively) establishes the mapping between the unary beliefs and the spatial distances. Different segmentations are achieved for different values of $\boldsymbol{\theta}$ via (6). Please refer to [20] for related work on (*non geodesic*) generalized distance transforms.

Figure 3 illustrates the effect of applying our filter (6) to *weak*, real-valued unaries. Larger values of θ not only tend to remove isolated islands (as illustrated earlier) but also produce smoother segmentation boundaries, in the presence of weak contrast and/or uncertain unaries. Furthermore, strong edges "lock" the segmentation in place. In summary, our filter produces segmentations which are smooth, edge-aligned and agree with the unaries. Thus the filter is ideally suited to be used for the generation of plausible segmentation hypotheses.

[3] In all experiments in this paper the value of μ is fixed to $\mu = 5$.

Energy minimization. We now search for the value $\boldsymbol{\theta}^*$ corresponding to the lowest energy $E_{GeoS} = E(\mathbf{z}, \boldsymbol{\alpha}(\boldsymbol{\theta}^*))$. For each value of $\boldsymbol{\theta}$ the segmentation operation in (6) requires 4 unsigned distance transforms. Thus, a naïve exhaustive search for $N_d \times N_e$ values of $\boldsymbol{\theta}$ would require 4 N_d N_e distance computations. However, it is easy to show that by pre-computing distances the load is reduced to only $2 + N_d + N_e$ operations [4], with an associated memory overhead. All of the above distance transforms are independent of each other and can be computed *in parallel* on appropriate hardware. Therefore, in a machine with N_c processors (cores) the total time T taken to run exhaustive search is $T = (2 + (N_d + N_e)/N_c)\, t$, with t the unit time required for each unsigned distance transform (9). An economical gradient descent optimization strategy may also be employed here. Comparative efficiency results are presented in section 5.

Selecting the search space. An important question at this point is how to choose the search space $\mathcal{S}$. As discussed earlier, $\boldsymbol{\theta}$ are intuitive parameters which represent the maximum size of the regions to be removed. Therefore, $\mathcal{S}$ must depend on the image resolution and on the spatial extent of noisy regions within the unary signal. Unless otherwise stated, for the approximately VGA-sized images used in this paper we have fixed $\mathcal{S} = \{5, 6, ..., 15\} \times \{5, 6, ..., 15\}$ (and thus $N_d = N_e = 10$).

Estimating the segmentation posterior. Computing the full CRF posterior $p(\boldsymbol{\alpha}) = 1/Z_p \ \exp\left(-E(\boldsymbol{\alpha})/\sigma_p\right)$ is impractical [5]. However, importance sampling [21] allows us to approximate $p(\boldsymbol{\alpha})$ with its Monte Carlo mean $\tilde{p}(\boldsymbol{\alpha})$. The *proposal distribution* $q(\boldsymbol{\alpha})$ can be computed as $q(\boldsymbol{\alpha}) = 1/Z_q \ \exp\left(-E(\boldsymbol{\alpha}(\boldsymbol{\theta}))/\sigma_q\right)$, $\forall \boldsymbol{\theta} \in \mathcal{S}$ (and $q(\boldsymbol{\alpha}) = 0 \ \forall \boldsymbol{\theta} \notin \mathcal{S}$). Then $\tilde{p}_N^q(\boldsymbol{\alpha}) = 1/n \sum_{i=1}^{n} p(\boldsymbol{\alpha}(\boldsymbol{\Theta}_i))/q(\boldsymbol{\alpha}(\boldsymbol{\Theta}_i))$, with the N samples $\boldsymbol{\Theta}_i$ generated from a uniform prior over $\mathcal{S}$. Since $\mathcal{S}$ is a small, quantized 2D space, in practice $\boldsymbol{\Theta}_i$ are generated deterministically by exploring the entire $\mathcal{S}$. The parameters σ_q, σ_p are trained discriminatively from hundreds of manually-labelled trimaps (e.g. fig. 4d). The estimated CRF posterior $\tilde{p}_N^q(\boldsymbol{\alpha})$ is used in fig. 4c"',d"' to compute the segmentation mean $\tilde{\boldsymbol{\alpha}} = \int_{\boldsymbol{\alpha}} \boldsymbol{\alpha} \tilde{p}_N^q(\boldsymbol{\alpha}) d\boldsymbol{\alpha}$ and the associated variance $\Lambda_{\boldsymbol{\alpha}}$. In interactive video segmentation, the quantity $\Lambda_{\boldsymbol{\alpha}}$ may for instance be used to detect unstable segmentations and ask the user to improve the appearance models by adding more strokes. Proposals sampled from $\mathcal{S}$ may also be fused together via QPBO [22].

Exploring more complex energy models. In contrast to graph-cut, here the energy and its minimization algorithm are decoupled. This fact is advantageous since now the choice of class of energies is no longer dominated by considerations of tractability. Our technique can thus be applied to more complex energy models than the one in (8). As an example, below we consider energies containing global terms:

$$E(\mathbf{z}, \boldsymbol{\alpha}) = U(\mathbf{z}, \boldsymbol{\alpha}) + \lambda V(\mathbf{z}, \boldsymbol{\alpha}) + \kappa G(\mathbf{z}, \boldsymbol{\alpha}) \tag{10}$$

The global soft constraint G cannot be written as a sum of unary and pairwise terms [23,24]. G captures global properties of image regions and can be used,

[4] The distance D_s need be computed only once per image as it does not depend on $\boldsymbol{\theta}$.

e.g. to encourage constraints on areas, global appearance, shape or context. For example in [23] $G = G(h_1, h_2)$ is defined as a divergence between region histograms h_i. General energy models of this kind have not been used much in the literature because of the lack of appropriate optimization techniques [25]. However, their usefulness is clear, and finding even approximate, efficient solutions is important. Results of this kind are presented in the next section.

5 Results and Applications

This section validates GeoS with respect to accuracy and efficiency. Qualitative and quantitative results on interactive and automatic image and video segmentation are presented.

Interactive image segmentation. Figure 4 shows a first example of interactive segmentation on a difficult standard test image showing camouflage [26]. The energy is defined as in (8). In this and all interactive segmentation examples, the unaries (fig. 4c) are obtained by: i) computing histograms over the RGB space quantized into 32^3 bins from the user provided strokes, and ii) evaluating the `Fg` and `Bg` likelihoods on all image pixels. As expected the GeoS MAP segmentation in fig. 4c' looks like a version of the unaries but with higher spatial smoothness of labels. The GeoS solution is very similar to the min-cut one (fig. 4c"). The segmentation mean $\tilde{\boldsymbol{\alpha}}$ and variance are also computed. The mean image $\tilde{\boldsymbol{\alpha}}$ can be thought of as an automatically computed trimap.

Computational efficiency. Here we compare the run times of GeoS and min-cut. For min-cut we use the public implementation in [28] and also our own implementation which has been optimized for grid graphs. GeoS has been implemented using SSE2 assembly instructions, exploiting cache efficiency and multi-threading for optimal performance. The data-level parallelism (SSE2) is made possible by noting that four of the five terms in the equation in fig. 12 are independent of the current scan-line. All experiments are run on an Intel Core2 Duo desktop with 3GB RAM and 2 × 2.6GHz CPU cores.

Figure 5 plots the run time curves obtained when segmenting the "llama" image as a function of image size. Both min-cut curves show a slightly "superlinear" behavior, while GeoS is linear. On a 1600 × 1200 image GeoS ($N_c = 4, N_d = N_e = 10$) produces a 12-fold speed-up with respect to min-cut. On-line video segmentation may be achieved by gradient descent because of the high temporal correlation of the energy in consecutive frames (*cf.* fig.5c, denoted "g.d."). Using 2 steps of gradient descent on 2 × 2 grids (typical values) produces a 21-fold speed-up. Geos' efficiency gain increases non-linearly for larger resolutions. For instance, on a $25Mpix$ image the GeoS ($N_c = 4, N_d = N_e = 10$) produces a 33-fold speed-up and gradient-descent GeoS a **60**-fold speed-up with respect to min-cut. Finally, while min-cut's run times depend on the quality of the unaries (the more uncertain, the slower the minimization) GeoS has a fixed running cost, thus making its behaviour more predictable.

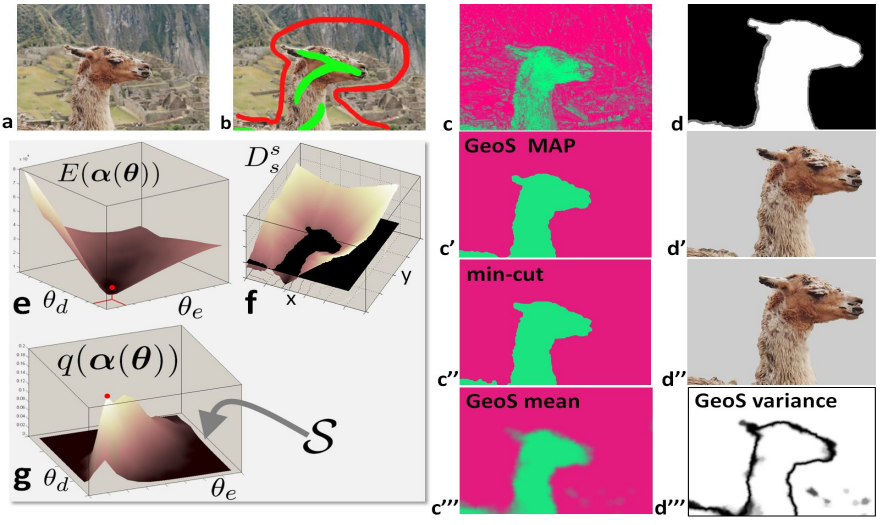


Fig. 4. GeoS v min-cut for interactive segmentation. **(a)** Input image.**(b)** user provided `Fg` and `Bg` strokes, **(c)** corresponding unaries (green for `Fg`, red for `Bg` and grey for uncertain). **(d)** ground truth segmentation (zoomed). **(e)** energy $E(\boldsymbol{\alpha}(\boldsymbol{\theta}))$, with the computed minimum marked in red. **(f)** The distance D_s^s corresponding to the optimum $\boldsymbol{\theta}^*$. **(g)** The proposal distribution $q(\boldsymbol{\alpha}(\boldsymbol{\theta}))$. **(c',d')** Resulting *GeoS MAP* segmentation $\boldsymbol{\alpha}^*$ and corresponding `Fg` layer. **(c",d")** *Min-cut* segmentation on the same energy. **(c"')** *GeoS mean* segmentation $\tilde{\boldsymbol{\alpha}}$, see text. Uncertain pixels are shown in grey. **(d"')** corresponding *GeoS variance* (dark for high variance).

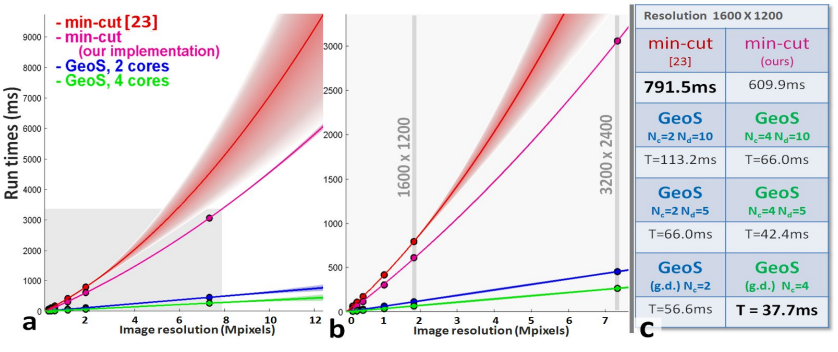
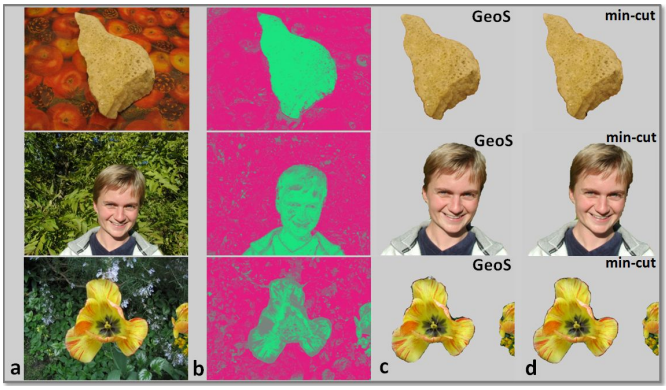


Fig. 5. Run time comparisons. **(a)** Run times for min-cut and GeoS for varying image size. Circles indicate our measurements. Associated uncertainties have been estimated by assuming Gaussian noise on the measurements [27]. Min-cut [28] fails to run on images larger than 1600×1200, thus yielding larger uncertainty for higher resolutions. **(b)** as in (a), zoomed into the highlighted region. Min-cut shows a slightly superlinear behaviour while GeoS is linear with a small slope. For large resolutions GeoS can be up to 60 times faster than min-cut. **(c)** Run-times for 1600×1200 resolution. Even for relatively low resolution images GeoS is considerably faster than min-cut. Identical energies are used for all four algorithms compared in this figure.



%	GeoS v min-cut	GeoS v gr.-truth	Min-cut v gr.-truth
sponge	0.22	0.21	0.18
person	0.23	0.51	0.44
flower	0.71	0.54	0.37
llama	0.87	0.62	0.32

Fig. 6. Interactive image segmentation. (a) Original test images: "sponge", "person" and "flower" from the standard test database in [8] (approx. VGA sized); **(b)** unaries computed from the user scribbles provided in [8]; **(c)** *GeoS* segmentations. **(d)** *min-cut* segmentations for the same energy as in (c), corresponding to a single iteration of GrabCut [26]. More iterations as in [26] help reduce the amount of manual interaction. **(e)** Percentage of differently classified pixels, see text.

When comparing GeoS with the algorithm in [29], GeoS yielded a roughly 30-fold speed-up factor while avoiding connectivity issues. Besides, the algorithm in [9,29] is designed for *interactive* segmentation only.

Segmentation accuracy. Figure 6 presents segmentation results on the standard test images used in [8]. To quantify the difference in segmentation quality between min-cut and GeoS we could use the relative difference between the minimum energy found, i.e. $\delta(E_{GeoS}, E_{min}) = (E_{GeoS} - E_{min})/E_{min}$, as in [8]. However, this is not a good measure since adding a constant term ΔE to the energy would not change the output segmentation while it would affect δ. Thus δ can be made very small by choosing a very large Δ. Here we chose to compare the GeoS and min-cut segmentations to each other and to the manually labelled ground truth by counting the number of differently classified pixels.

Results for the four example images encountered so far are reported in fig. 6e. In each case the optimum value of λ (learned discriminatively for min-cut) was used. The min-cut and GeoS results are very close visually and quantitatively, with the number of differently labelled pixels well below 1% of the image area. The largest difference is for the "llama" image where the furry outline makes both solutions equally likely. All segmentations are also very close to the ground truth. The three GeoS segmentations in fig. 6c were obtained in $< 10ms$ each (to be compared with the much larger timings reported in [8]).

Contrast sensitivity. In fig. 7 contrast-sensitivity enables thin protrusions to be segmented correctly, despite the absence of flux in the energy. Contrast is especially important with weaker unaries such as shown in fig. 3 and fig. 8. In fig. 8b,b', using patches to compute stereo likelihoods [3] causes their misalignment with respect to the foreground boundary. Using $\gamma > 0$ in GeoS encourages

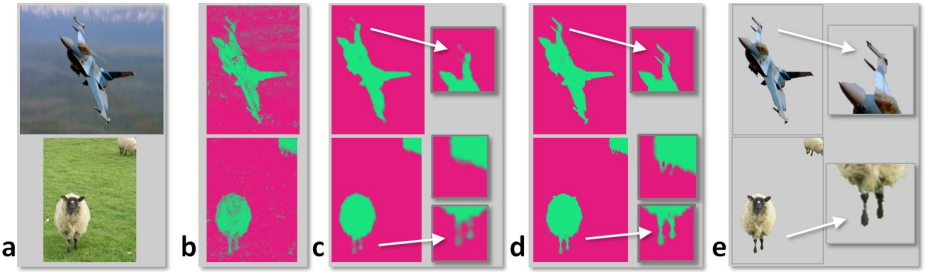


Fig. 7. The effect of contrast on thin structures. (a) Two input images. **(b)** unaries (from user strokes); **(c)** GeoS mean segmentation with no contrast. The smoothness prior makes thin protrusions (e.g. the sheep legs or the planes missiles) uncertain (grey). **(d)** GeoS mean segmentation with contrast enabled. Now the contrast-sensitive pairwise term correctly pulls the aeroplane and sheep thin protrusions in the foreground. **(e)** GeoS MAP segmentation for the contrast-sensitive energy in (d).

Fig. 8. Segmentation results in the presence of weak, stereo unaries. (a,a') Frames from two stereo videos. **(b,b')** Stereo likelihoods, with large uncertain areas (in grey). **(c,c')** GeoS segmentation, with no contrast sensitivity. **(d,d')** As in (c,c') but with contrast sensitivity. Now the segmentation accurately follows the person's outline.

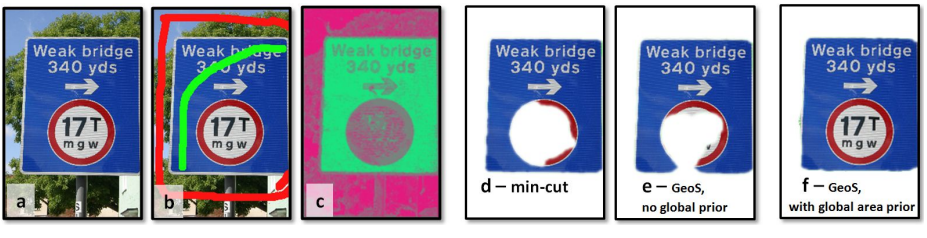


Fig. 9. Exploiting global constraints. (a) Original test image; **(b)** user provided `Fg` and `Bg` strokes; **(c)** corresponding unaries; **(d)** min-cut segmentation with $\kappa = 0$; The circular traffic sign is missed out. **(e)** GeoS segmentation on same energy as in (d); **(f)** GeoS segmentation on energy with global constraint $G = |Area_{\mathtt{Fg}}/Area - 0.7|$.

the segmentation to follow the person's silhouette correctly (fig. 8d,d'). Next we experiment with more complex energies, containing global terms.

Exploiting global energy constraints. The example in fig. 9 shows the effect of the global constraint G in (10). Energies of the kind in (10) cannot be minimized by min-cut. In the segmentations in fig. 9d,e (where $\kappa = 0$) the circular weight limit sign is missed. This problem is corrected in fig. 9f which uses the energy (10) (with $k > 0$). The additional global term G is defined

as $G = |Area_{\mathtt{Fg}}/Area - 0.7|$ to encourage the Fg region to cover about 70% of the image area. Similar results are obtained on this image by imposing soft constraints on global statistics of appearance or shape (see also [30]).

Segmenting n-dimensional data. Geodesic transforms and thus GeoS can easily be extended to more than 2 dimensions. Figure 10 shows an example of segmentation of the space-time volume defined by a time-lapse video of a growing flower. Figure 11 shows segmentation of 3D MRI data. In each case brush strokes applied in two frames suffice to define good unaries. Individual organs are highlighted in fig. 11 by repeated segmentation (see also [31]).

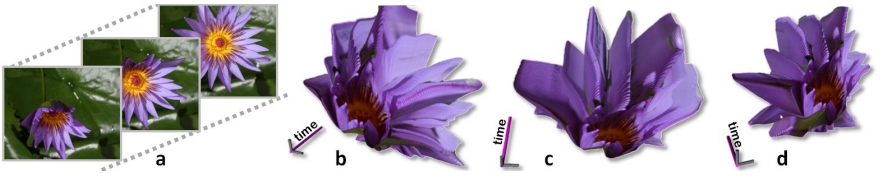


Fig. 10. Batch, space-time segmentation of video. (a) Three frames of a time-lapse video of a growing flower. **(b, c, d)** Three views of the segmented "video-cube". GeoS segmentation is achieved directly in the space-time volume of the video.

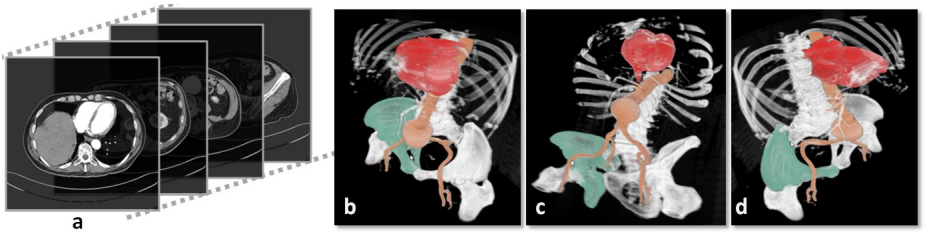


Fig. 11. Segmentation of 3D, medical data. (a) Some of the 294 512×512 input grey-scale slices from a patient's torso. **(b,c,d)** GeoS segmentation results. Bones, heart and aorta have been accurately separated from the remaining soft tissue, directly in the 3D volume. Different organs have been coloured to aid visual inspection.

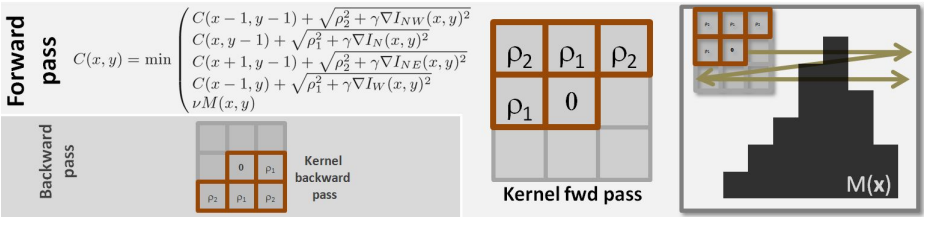


Fig. 12. Efficient geodesic distance transform. See appendix.

6 Conclusion

This paper has presented GeoS, a new algorithm for the efficient segmentation of n-D images and videos. The key contribution is an approximate energy minimization technique which finds the segmentation solution by economical search within a restricted space. Such space is populated by good, spatially-smooth, contrast-sensitive solution hypotheses generated by our new, efficient geodesic operator. The algorithm's reduced search space, contiguous memory access and intrinsic parallelism account for its efficiency even for high resolution data.

Extensive comparisons between GeoS and min-cut show comparable accuracy; with GeoS running many times faster and being able to handle more general energies.

References

1. Boykov, J., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-D images. In: IEEE ICCV (2001)
2. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. In: ACM Trans. on Graphics (SIGGRAPH) (2004)
3. Kolmogorov, V., Criminisi, A., Blake, A., Cross, G., Rother, C.: Bilayer segmentation of binocular stereo video. In: IEEE CVPR (2005)
4. Criminisi, A., Cross, G., Blake, A., Kolmogorov, V.: Bilayer segmentation of live video. In: IEEE CVPR (2006)
5. Kohli, P., Torr, P.H.S.: Measuring uncertainty in Graph Cut solutions. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 30–43. Springer, Heidelberg (2006)
6. Kolmogorov, V., Zabih, R.: Multi-camera scene reconstruction via graph cuts. In: ECCV (2002)
7. Sinop, A.K., Grady, L.: A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In: IEEE ICCV (2007)
8. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., Rother, C.: A comparative study of energy minimization methods for markov random fields. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 16–29. Springer, Heidelberg (2006)
9. Bai, X., Sapiro, G.: A geodesic framework for fast interactive image and video segmentation and matting. In: IEEE ICCV (2007)
10. Komodakis, N., Tziritas, G., Paragios, N.: Fast, approximately optimal solutions for single and dynamic MRFs. In: IEEE CVPR (2007)
11. Lempitsky, V., Rother, C., Blake, A.: Logcut - efficient graph cut optimization for markov random fields. In: IEEE ICCV, Rio (2007)
12. Juan, O., Boykov, J.: Active graph cuts. In: IEEE CVPR (2006)
13. Kohli, P., Torr, P.: Dynamic graph cuts for efficient inference in markov random fields. PAMI (2007)
14. Fabbri, R., Costa, L., Torrelli, J., Bruno, O.: 2d euclidean distance transform algorithms: A comparative survey. ACM Computing Surveys 40 (2008)
15. Jones, M., Baerentzen, J., Sramek, M.: 3d distance fields: a survey of techniques and applications. IEEE Trans. on Visualization and Computer Graphics 12 (2006)
16. Borgefors, G.: Distance transformations in digital images. Computer Vision, Graphics and Image Processing (1986)

17. Sethian, J.A.: Fast marching methods. SIAM Rev. 41 (1999)
18. Toivanen, P.J.: New geodesic distance transforms for gray-scale images. Pattern Recognition Letters 17, 437–450 (1996)
19. Yatziv, L., Bartesaghi, A., Sapiro, G.: O(n) implementation of the fast marching algorithm. Journal of Computational Physics 212, 393–399 (2006)
20. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. IJCV 61 (2005)
21. Ripley, B.D.: Stochastic Simulation. Wiley and Sons, Chichester (1987)
22. Rother, C., Kolmogorov, V., Lempitsky, V.T., Szummer, M.: Optimizing binary MRFs via extended roof duality. In: IEEE CVPR (2007)
23. Rother, C., Kolmogorov, V., Minka, T., Blake, A.: Cosegmentation of image pairs by histogram matching - incor. a global constraint into MRFs. In: CVPR (2006)
24. Kolmogorov, V., Boykov, J., Rother, C.: Applications of parametric maxflow in computer vision. In: IEEE ICCV, Rio (2007)
25. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE Trans. PAMI 26 (2004)
26. Blake, A., Rother, C., Brown, M., Perez, P., Torr, P.: Interactive image segmentation using an adaptive GMMRF model. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 428–441. Springer, Heidelberg (2004)
27. Bishop, C.M.: Pattern Recognition and machine Learning. Springer, Heidelberg (2006)
28. `http://www.adastral.ucl.ac.uk/~vladkolm`
29. Bai, X., Sapiro, G.: A geodesic framework for fast interactive image and video segmentation and matting. Technical Report 2185, Institute of Mathematics and Its Applications, Univ. Minnesota Preprint Series(2008)
30. Cremers, D., Osher, S.J., Soatto, S.: Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. IJCV 69 (2006)
31. Yatziv, L., Sapiro, G.: Fast image and video colorization using chrominance blending. IEEE Trans. on Image Processing 15 (2006)

Appendix – Fast Geodesic Distance Transform

Given a map $M(\mathbf{x}) \in [0, 1]$, in the forward pass the map is scanned with a 3×3 kernel from the top-left to the bottom-right corner and the intermediate function $C(\mathbf{x})$ is iteratively constructed as illustrated in fig. 12. The north-west, north, north-east and west components of the image gradient ∇I are used. The ρ_1 and ρ_2 local distances are usually set to $\rho_1 = 1$ and $\rho_2 = \sqrt{2}$. In the backward pass the algorithm proceeds from the bottom-right to the top-left corner and applies the backward kernel to $C(\mathbf{x})$ to produce the final distance $D(\mathbf{x})$ (*cf.* fig. 1). Larger kernels produce better approximations to the exact distance.

Simultaneous Motion Detection and Background Reconstruction with a Mixed-State Conditional Markov Random Field

Tomás Crivelli[1,2], Gwenaelle Piriou[3], Patrick Bouthemy[2], Bruno Cernuschi-Frías[1,2], and Jian-feng Yao[2,4]

[1] University of Buenos Aires, Buenos Aires, Argentina
[2] INRIA Rennes, Irisa, France
[3] Université de Bretagne-Sud, Vannes, France
[4] IRMAR, Rennes, France
tcrivell@irisa.fr

Abstract. We consider the problem of motion detection by background subtraction. An accurate estimation of the background is only possible if we locate the moving objects; meanwhile, a correct motion detection is achieved if we have a good available background model. This work proposes a new direction in the way such problems are considered. The main idea is to formulate this class of problem as a joint decision-estimation unique step. The goal is to exploit the way two processes interact, even if they are of a dissimilar nature (symbolic-continuous), by means of a recently introduced framework called mixed-state Markov random fields. In this paper, we will describe the theory behind such a novel statistical framework, that subsequently will allows us to formulate the specific joint problem of motion detection and background reconstruction. Experiments on real sequences and comparisons with existing methods will give a significant support to our approach. Further implications for video sequence inpainting will be also discussed.

1 Introduction

The recent advances in computer vision have been moving towards the quest for the development of systems and algorithms able to tackle complex situations where an integrated and optimal decision-estimation process is required. Efficient early vision techniques are nowadays able to feed subsequent stages of high-level information processing in a desirable way: fast, accurate and robust. Anyway, there has been always a component of sequentiality that tends to address a certain task as a succession of atomic steps. Consider the problem of foreground moving objects detection by background subtraction, where a model of the background or reference image is usually learned and motion detection is solved from differences between image observations and such a model. A "chicken-and-egg" situation arises when we want to set an optimal approach for both tasks: an accurate estimation of the background is only possible if we know which regions of the image belong to it, that is, if we locate the moving objects; meanwhile,

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 113–126, 2008.

a correct motion detection is achieved if we have a good available background model.

This work proposes a new direction in the way such problems are considered. The main idea is to formulate a unique and joint decision-estimation step, which means more than simply solving two (or more) problems at the same time (either sequentially, iteratively or adaptively). We emphasize that the goal is to exploit the way two processes interact, even if they are of a dissimilar nature.

Returning to the problem of motion detection and background modeling, we can redefine the problem as an example of the starting point for our proposal: let us consider that a point in the image is a single process that can take two types of values, a symbolic value (or abstract label) accounting for a positive motion detection, or a numeric value associated to the brightness intensity of the reference image at that location. Consequently, what would it mean to solve both tasks jointly in this context, is to obtain a single optimal estimate of such a process (Fig. 1).

Additionally, our method relies not only on the comparison between the current image and the reference image but explicitly integrates motion measurements obtained between consecutive images. Conditional random fields [1], extended to a mixed-state version, allow us to introduce these observations (or any other) in the model and contributes to make the overall scheme complete, accurate and powerful.

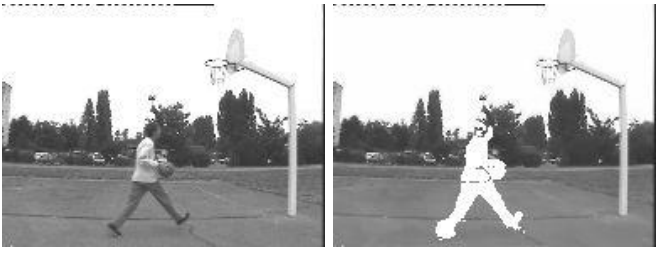


Fig. 1. Left: original image from the Basketball sequence. Right: a mixed-state field obtained with the proposed motion detection method. In white it is represented the symbolic part, accounting for a positively detected moving point. The continuous part is represented by the reconstructed background

This paper is organized as follows. In section 2, we will sufficiently describe the theory behind such a novel statistical framework. In section 3, a review of motion detection by background subtraction techniques with their advantages and drawbacks are discussed. Section 4 is devoted to the formulation of the proposed conditional mixed-state model for simultaneous motion detection and background reconstruction. In section 5 we show experiments on real sequences and comparisons with existing methods, which will give a significant support to our approach. These comparisons will show an improvement in the detection rate, diminishing the number of false positives and negatives, together with a correct reconstruction of the real background image, not a model of it. Further

implications and results of this method for video sequence inpainting will be also discussed.

2 The Mixed-State Statistical Framework and Related Approaches

The concept of a random process that can take different types of values (either numerical or abstract) according to a generalized probability function, is formalized through the so-called *mixed-state random variables*. This includes diverse situations. We have formulated a mixed discrete-continuous Markov random field in [2] in the context of modeling of dynamic or motion textures. In this case, it is demonstrated that normal flow scalar motion observations arised from these types of sequences show a discretely distributed value at zero (null-motion) and a Gaussian-like continuous distribution for the rest of the values. This model was extended in [3] and applied to the problem of motion texture segmentation. Previously, Salzenstein and Pieczynski [4] (more recently in [5]), have proposed a fuzzy image segmentation model where the fuzzy labels are a particular instance of mixed-state variables with values in $[0, 1]$.

Our work takes a step further, not in the theoretical aspect of the framework, but in the exploitation of its implications in computer vision.

Let us define $\mathcal{M} = \{\omega\} \cup \mathbb{R}$, with ω a "discrete" value, called *symbolic* value. A random variable X defined on this space, called *mixed-state variable*, is constructed as follows: with probability $\rho \in (0, 1)$, set $X = \omega$, and with probability $1 - \rho$, X is continuously distributed in $\mathbb{R}$. In order to compute the probability density function of X, $\mathcal{M}$ is equipped with a "mixed" reference measure $m(dx) = \nu_\omega(dx) + \lambda(dx)$, where ν_ω is the discrete measure for the value ω and λ the Lebesgue measure on $\mathbb{R}$. Let us define the indicator function of the symbolic value ω as $\mathbf{1}_\omega(x)$ and its complementary function $\mathbf{1}^*_\omega(x) = \mathbf{1}_{\{\omega\}^c}(x) = 1 - \mathbf{1}_\omega(x)$. Then, the above random variable X has the following density function w.r.t. $m(dx)$:

$$p(x) = \rho \mathbf{1}_\omega(x) + (1 - \rho)\mathbf{1}^*_\omega(x)p^c(x), \tag{1}$$

where $p^c(x)$ is a continuous pdf w.r.t. λ, defined on $\mathbb{R}$. Hereafter, such generalized density will be called *mixed-state density*.

2.1 Mixed-State Markov Models

In the context of Markov fields, the concept of mixed-state random variables and mixed-state densities, derives in the definition of mixed-state conditional densities. Let $S = \{1....N\}$ be a lattice of points or image locations such that $\mathbf{X} = \{x_i\}_{i \in S}$. Define $\mathbf{X}_A$ as the subset of random variables restricted to $A \subset S$, i.e., $\mathbf{X}_A = \{x_i\}_{i \in A}$. Then we write:

$$p(x_i \mid \mathbf{X}_{\mathcal{N}_i}) = \rho(\mathbf{X}_{\mathcal{N}_i})\mathbf{1}_\omega(x_i) + (1 - \rho(\mathbf{X}_{\mathcal{N}_i}))\mathbf{1}^*_\omega(x_i)p^c(x_i \mid \mathbf{X}_{\mathcal{N}_i}), \tag{2}$$

where $\rho(\mathbf{X}_{\mathcal{N}_i}) = P(x_i = \omega \mid \mathbf{X}_{\mathcal{N}_i})$ and $\mathbf{X}_{\mathcal{N}_i}$ is the subset of $\mathbf{X}$ restricted to a neighborhood of locations $\mathcal{N}_i$. Equation (2) defines the local characteristics of a global random field, that will respond to a nearest neighbor Gibbs distribution, as stated by the equivalence of Hammersley-Clifford [6]. Moreover, it is the starting point for defining a model that allows to obtain a regularized symbolic-continuous field. We recall one useful result of the proposed statistical framework (see [6,7]) for the case of second order Markov random fields:

Result 1. *For a second order Markov random field that responds to a family of conditional densities given by (2) the associated joint Gibbs distribution* $Z^{-1} \exp Q(\mathbf{X}) = Z^{-1} \exp -H(\mathbf{X})$ *is given by the energy:*

$$H(\mathbf{X}) = \sum_{i \in S} \left\{ V_i^d(x_i) + V_i^c(x_i) \right\} + \sum_{<i,j> \in S} \left\{ V_{i,j}^d(x_i, x_j) + V_{i,j}^c(x_i, x_j) \right\}, \qquad (3)$$

where $V_i^d(x_i) = \alpha_i \mathbf{1}_\omega^*(x_i)$ *and* $V_{i,j}^d(x_i, x_j) = \beta_{ij} \mathbf{1}_\omega^*(x_i) \mathbf{1}_\omega^*(x_j)$*, that is they correspond to purely discrete potentials, and* $V_i^c(x_i)$ *and* $V_{i,j}^c(x_i, x_j)$ *are energy terms related to the continuous part* p^c*.*

Thus, we know the general shape of the potentials for a mixed-state model. In what follows, we apply this result to the formulation of the joint problem of motion detection and background reconstruction as a conditional mixed-state Markov random field estimation problem.

3 Motion Detection by Background Subtraction: Overview of Existing Methods

One of the most widely used methods for motion detection is *background subtraction*. The approach, derived initially from a thresholding process over the difference between the observed intensity (or color) at a point and a reference value representing the background, has evolved into more complex schemes where the shared idea is to consider that a foreground moving object does not respond to some representation of the background.

For existing methods, a necessary step consists in the learning of the background and this implies either the availability of training frames with no moving objects, or the assumption that a point belongs to the background most of the time. Adaptive schemes have also been proposed in order to update the model sequentially and selectively, according to the result of the detection step. Anyway, a general consensus has been to estimate a probability density for each background pixel. The simplest approach is to assume a single Gaussian per pixel (see for example [8]), whose parameters may be estimated by simple running averages or even median filters. A valid and certain criticism to this hypothesis is that the distribution of the intensity of a background pixel over time can vary considerably, but usually repeatedly. In that direction, multi-modal density models seemed to perform better. Mixtures of Gaussians [9] and non-parametric models [10,11,12] have shown good results, able to deal with the dynamic of the background distribution.

However they suffer from several drawbacks. The approach does not assume spatial correlation between pixels, nor in the model of the background, neither in the binary detection map. Aware of this, posterior morphological operations are applied in order to achieve some sort of regularization in the resulting motion detection map. No regularization is proposed for the reference model.

They need also to incorporate points detected as foreground to estimate the background model (called blind update) in order to avoid deadlock situations, where a badly estimated background value for a pixel results in a continuously and wrongly detected moving point. This leads to bad detections as intensity values that do not belong to the background are incorporated to the model. A lot of heuristic corrections are usually applied in order to correct this drawback, but unfortunately, introducing others. Finally, they are very sensitive to the initialization of the background model, particularly, when an initial image with no moving objects is not available in the video sequence.

The advantages of incorporating spatial context and regularization, in the background and also the foreground, are demonstrated for example in [13,14] by means of a Markov random field model and ARMA processes, respectively. As for another energy-based method for background subtraction, the work of Sun et al. [15] on Object Cut, models the likelihood of each pixel belonging to foreground or background along with an improved spatial contrast term. The method relies on a known (previously learned) background model and an adaptive update scheme is necessary. Finally in [16], a technique for motion detection, not based on background modeling, but on clustering and segmentation of motion and photometric features, is described, where explicit spatial regularization is introduced through a MAP-MRF approach.

3.1 Our Method

Based on these observations we propose a simultaneous motion detection and background reconstruction method with the following characteristics:

- **Reduction of false positive and false negatives.** Through a more complex regularization of the detection map, exploiting spatial priors, and the interaction between symbolic and continuous states.
- **Reconstruction of the background.** Obtaining a reconstructed reference image, not just a model of it, will allows to exploit the local information of the difference between the background and a foreground moving object, avoiding the undesirable effects of modeling noise, which is filtered out from the reconstructed image.
- **No need of training samples.** Through a temporal update strategy which can be adopted thanks to a correct regularized estimation of the motion map, the reference image is reconstructed on-the-fly on those regions temporally not occluded by the moving objects.
- **Joint decision-estimation solution.** Exploiting simultaneously the information that the reference image provides for motion detection, and vice versa.

4 A Conditional Mixed-State Model for Motion Detection

4.1 Definitions

Let us call $\mathbf{y}_t = \{y_i^t\}_{i \in S}$ the intensity image at time t, where $y_i^t \in [0, 255]$ is the brightness intensity value at location $i \in S = \{1....N\}$ of the image grid. Then $\mathbf{y} = \{\mathbf{y}_t\}_t$ is a sequence of images that we call *observations*. We will associate a positive motion detection for a single point to the abstract label ω. Then, we define a mixed-state random field $\mathbf{x}_t = \{x_i^t\}_{i \in S}$ where $x_i^t \in \mathcal{M} = \{\omega\} \cup [0, 255]$ is a mixed-state random variable.

Suppose we have an estimate of $\mathbf{x}_t$ for a given instant t, that is, the moving points and the estimated intensity value for the background at the non-moving points. We can use this information and the past estimated $\mathbf{x}_{t'}$ (for $t' < t$) to reconstruct the reference image at t, that we call $\mathbf{z}_t = \{z_i^t\}_{i \in S}$. We propose to update the background estimation as follows,

$$z_i^t = \begin{cases} x_i^t & \text{if } x_i^t \neq \omega \\ z_i^{t-1} & \text{otherwise.} \end{cases} \tag{4}$$

The rationale of this rule is that when we do not detect motion, we have a good estimation for the reference intensity value at a given point, so we can use this value as a background value; as the objects in the scene move, we can progressively estimate the background for different parts of the image. In other words, we can fill the gaps at those moments where the background is not occluded.

4.2 Energy Terms

Let us call $H(\mathbf{x}_t \mid \mathbf{y}, \mathbf{z}_{t-1})$ the energy function to be minimized, associated to a *conditional* mixed-state Markov random field as proposed in (3), given the observations $\mathbf{y}$ and the previously available background image [1]. In what follows we design the mixed-state energy terms. Considering a *conditional* Markov random field, as introduced in [1], allows us to define these energy terms in a flexible way, in particular it enables to exploit a large set of observations (e.g., a block) at each site. That is, it is able to integrate at an image location any information extracted from the input data and obtained across arbitrary spatial or temporal (or both) neighborhoods, or information from previously reconstructed variables, or even the association of both.

We will consider three types of energy terms. The *discriminative* term, which plays a role in the decision process, penalizing or favoring the presence of motion for a point given the observations; the *reconstruction* terms, involved in the estimation of the reference image, which also affects the motion detection decision process by means of background subtraction; and the *regularization* terms, related to the smoothing of the mixed-state field.

[1] The extension of the previous stated results to a mixed-state model conditioned to an observation process is straightforward.

First we propose to introduce a discriminative term related to the symbolic part of the field, that is, the motion detection map. Thus, we define a first-order potential

$$V_i^D(x_i^t \mid \mathbf{y}) = \alpha_i^D(\mathbf{y})\mathbf{1}_\omega^*(x_i^t), \tag{5}$$

where the weight $\alpha_i^D(\mathbf{y})$ depends on the observations and aims at tuning the belief of motion for a point. We propose to use $\alpha_i^D(\mathbf{y}) = -\log NFA_i(\mathbf{y}_{t-1}, \mathbf{y}_t, \mathbf{y}_{t+1})$, where $NFA_i(\cdot)$ stands for the *Number of False Alarms* obtained through an a contrario decision framework as in [17]. Its value is computed using three consecutive frames and taking the magnitude of the local normal flows, and constitutes a measure of the belief that a point belongs to the background (or conversely, to moving objects). We have implemented the simplest scheme proposed in [17], considering detection over square regions, usually of size 20x20 at each site i. A small value of NFA indicates a large belief of motion and conversely. Consequently, a low value of $\log NFA_i$ favors $x_i^t = \omega$ (Fig. 2). Thus, our method relies not only on the comparison between the current image and the reference image but explicitly introduces motion measurements. The overall scheme gains accuracy and completeness, integrating this low-level feature in the decision process. The flexibility of the conditional random field formulation [1] allows us to exploit these observations within the mixed-state model.

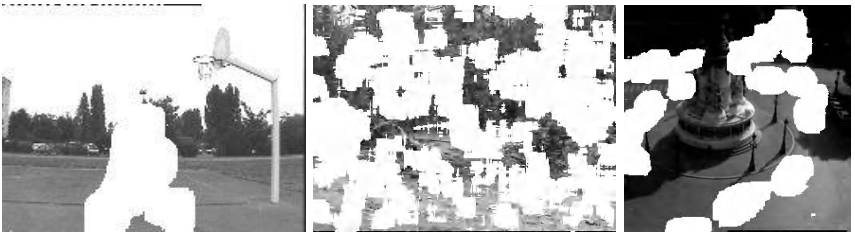


Fig. 2. Initial motion detection by computing the Number of False Alarms. The motion map is obtained by thresholding this quantity as explained in [17]. From left to right: the results are shown for the Basketball sequence (see Fig. 3), the Forest sequence (see Fig. 4) and the Traffic Circle sequence (see Fig. 5). Note that this quantity, with the basic implementation utilized here, over-regularizes the detection map, as it is a block-based detection strategy.

We elaborate now the reconstruction potential. On one side, it aims at estimating the intensity values of the background (reference) image, and exploits the flexibility of the mixed-state model in taking into account their interactions with the symbolic values. On the other side, here is where we introduce the term that compares the current image with the reconstructed reference image, which provides the basis for the decision process in a background subtraction method. We then write

$$V_i^R(x_i^t \mid \mathbf{y}, \mathbf{z}_{t-1}) = \gamma \left[\mathbf{1}_\omega^*(x_i^t) \frac{\left[x_i^t - m(z_i^{t-1}, y_i^t)\right]^2}{\sigma_i^2} + \mathbf{1}_\omega(x_i^t)\alpha_i^R(\mathbf{y}_t, \mathbf{z}_{t-1}) \right]. \tag{6}$$

First, we set $m(z_i^{t-1}, y_i^t) = cz_i^{t-1} + (1-c)y_i^t$ if we have an available previously estimated value for the reference image at that point, or $m(z_i^{t-1}, y_i^t) = y_i^t$ otherwise. Thus, the first term favors that, when there is no motion, i.e. $\mathbf{1}_\omega^*(x_i^t) = 1$, the estimated intensity value for a point is close to the previous estimated reference image, and simultaneously, penalizes the absence of motion if this difference is eventually large. Both types of values interact consequently, in order to minimize the energy. Note that this term also performs a temporal regularization of the reference estimates z_i^t by the interpolation form of the $m(\cdot)$ function. Furthermore, it is normalized by a local variance σ_i^2 estimated locally from $\mathbf{y}_t$. In the second term, we set,

$$\alpha_i^R(\mathbf{y}_t, \mathbf{z}_{t-1}) = \sigma_i^2 \left[n^{-1} \sum_{j \in \mathcal{N}_i} (z_j^{t-1} - y_j^t) \right]^{-2}, \tag{7}$$

resulting in a penalization of the presence of motion when the difference of intensity between the observation and the reference image is small. A local average of differences is introduced in order to reduce the effect of the noise present in the observations.

The potentials introduced so far, are in fact first-order terms, that relate the random variable at a point i w.r.t. the observations. Next, we introduce terms related to the regularization of the field. The objective is to have connected regions for the motion detection map, and a reconstructed background with a reduced amount of noise, but conserving edges and contrast of the image. Then, we add the following second-order mixed potential,

$$V_{ij}^S(x_{i,t}, x_{j,t} \mid \mathbf{y}) = \frac{\beta^c}{g_i(\nabla \mathbf{y}_t)} \mathbf{1}_\omega^*(x_i^t) \mathbf{1}_\omega^*(x_j^t) \left[\frac{(x_i^t - x_j^t)^2 - K}{\sigma_i^2} \right] - \frac{\beta^m}{g_i(\nabla \mathbf{y}_t)} \mathbf{1}_\omega(x_i^t) \mathbf{1}_\omega(x_j^t), \tag{8}$$

where $g_i(\nabla \mathbf{y}_t) = \max(1, \| \nabla y_i^t \|^2)$. A combined spatial regularization of both types of values is achieved through this energy potential. First, a Gaussian continuous term is introduced in order to obtain homogeneous intensity regions for the objects in the background. This regularization is only done when both points are not in motion and is stronger for those points where the image gradient is small, in such a way that we avoid the blurring of edges. Then, regarding the motion detection map[2], we observe that the amount of regularization depends as well on the continuous part, that is, is favored in homogeneous intensity regions. The constant K is set to the value $K = \frac{1}{2}(x_{\max} - x_{\min})^2 = (255)^2/2$, centering the range of values for this term and is introduced in order to indeed favor this regularization when two neighboring points tend to have similar intensities. If $K = 0$, the whole term can become null in that case, suppressing the regularization between adjacent points over non-moving regions. Another term for the smoothness of the moving points is added as well, in order to improve regularization and reduce false negative detections.

[2] More precisely, its complement, the non-motion map.

4.3 Estimation

The complete expression for the energy is finally,

$$H(\mathbf{x}_t \mid \mathbf{y}, \mathbf{z}_{t-1}) = \sum_i \left\{ V_i^D(x_i^t \mid \mathbf{y}) + V_i^R(x_i^t \mid \mathbf{y}, \mathbf{z}_{t-1}) \right\} + \sum_{i,j} V_{ij}^S(x_i^t, x_j^t \mid \mathbf{y}), \quad (9)$$

and the problem reduces to the task of estimating the field $\mathbf{x}_t$ by minimizing H. The ICM (Iterated Conditioned Modes) algorithm is applied. The concept of iteratively maximizing the conditional densities is equally applicable to generalized mixed-state densities as (2) and is equivalent to the minimization of an energy function $H(\cdot)$. Then, for each point the following rule is applied:

$$x_i^t = \begin{cases} \omega & \text{if } H(x_i^t = \omega \mid \mathbf{X}_{\mathcal{N}_i}, \mathbf{y}) < H(x_i^t = x_i^* \mid \mathbf{X}_{\mathcal{N}_i}, \mathbf{y}) \\ x_i^* & \text{otherwise} \end{cases} \quad (10)$$

where $H(x_i^t \mid \mathbf{X}_{\mathcal{N}_i}, \mathbf{y})$ is the energy associated to the conditional mixed-state density obtained from (9) and x_i^* is the continuous value that minimizes its continuous part, i.e. when $x \neq \omega$, here equals:

$$x_i^* = \frac{\frac{\beta^c}{g_i(\nabla \mathbf{y}_t)} \sum_{i,j} x_j^t \mathbf{1}_\omega^*(x_j^t) + \gamma \alpha_i^R(\mathbf{y}_t, \mathbf{z}_{t-1})}{\frac{\beta^c}{g_i(\nabla \mathbf{y}_t)} \sum_{i,j} \mathbf{1}_\omega^*(x_j^t) + \gamma}. \quad (11)$$

Note that this value is in fact, the mean of the conditional continuous density in (2), that results to be Gaussian, and is the estimated value for the reference image at point i.

5 Results and Experimental Comparisons

We have applied our method to real sequences consisting of articulated and rigid motion. As well, we compare the results with the methods of Stauffer and Grimson [9] and Elgammal et al. [10], for which we obtained an implementation from `http://www.cs.ucf.edu/~jdever/code/scode.html` and `http://cvlab.epfl.ch/~tola/source_code.html` respectively. At the same time, we compare the performance of the full mixed-state model, with two sequential implementations based on simplified (non-mixed) versions of the proposed energy potentials, in order to show the importance of the mixed-state terms. Firstly, we have implemented a sequential algorithm using equation (5) and the second term of equation (6): the first step is to estimate the moving points and then, with a fixed detection map, the background is updated. No regularization is introduced, nor in the detection or the background reconstruction. Secondly, we have implemented another sequential algorithm, now including non-mixed regularization, that is, using the potential of equation (5), the second term of equation (6) and the second term of equation (8). In other words, we take out the mixed potentials from the energy.

For our method, we use the 8-point nearest neighbor set, as the neighborhood for the mixed-state Markov random field. The parameters of the model were set

as following: $\gamma = 8$, $\beta^c = 1$, $\beta_m = 5$ and $c = 0.7$. For all the sequences these same values were used. This is justified observing equation (11). Assume all neighbor points are not in motion, then the estimated value for the background intensity is a weighted average between the 8 neighbors and the previous estimated background. Setting $\beta^c = 1$ we get a total weight of 8 for the surrounding points (if the local gradient is small), and then with $\gamma = 8$, we give the same weight to the previous estimated value. This situation establishes an equilibrium working point of the algorithm, from which we derived the order of magnitude of the parameters. β^m was set empirically in order to effectively remove isolated points. Anyway, the results practically did not show variations for $\beta^c \in [0.5, 1]$, $\gamma \in [8, 12]$ and $\beta^m \in [3, 6]$. This low sensitivity allowed us to fix a unique set of parameters for all the samples.

In Fig. 3 we show the result of comparing the different algorithms applied to the Basketball sequence. The method by Stauffer and Grimson shows false positively detected moving points in the background. The method by Elgammal et al. performs better, but has some problems at correctly achieving connected regions. The mixed-state method shows an improved regularization of the motion map, visually reducing false positives and false negatives, also compared with the sequential non-mixed versions of the algorithm.

Fig. 4 shows a complex scene of two man walking through a forest. In this example the background is not completely static as there is swaying vegetation. Our method supplies the best results discarding practically all the background motion, even compared with multi-modal density models. The proposed observations (Number of False Alarms) introduced in the discriminative term are able to cope with this kind of background dynamics.

Finally in Fig. 5 we show the result for a sequence of a Traffic Circle with multiple rigid motions. In this case, never during the sequence a complete background image is available. The cars continuously pass around the square entering and leaving the scene. The method by Stauffer and Grimson shows a deadlock situation due to the lack of training samples: initially the algorithm includes in the background some of the moving cars, resulting in a continuous positive wrong detection for subsequent frames and takes too long for the model to finally remove them from the reference image. Moreover, some regions of the background are never correctly updated. The same sequence tested with the non-parametric method of Elgammal et al. failed in generating valid results, resulting in an everywhere negative detection for mostly every point and every frame. The lack of training samples for the background, on which the method relies, is likely to be the cause of the failure.

For the method proposed here these problems are not present. The cars are well detected with less false positives for the mixed-state method. The algorithm is not able to distinguish the small cars entering the scene from the street in the top, grouping all in a single connected region. In this case, the separation between the cars in that region is about 4 pixels (the image is of size 256x256), which is in the order of the size of the considered neighborhoods used in the regularization terms. Nevertheless, it results in a well segmented scene where

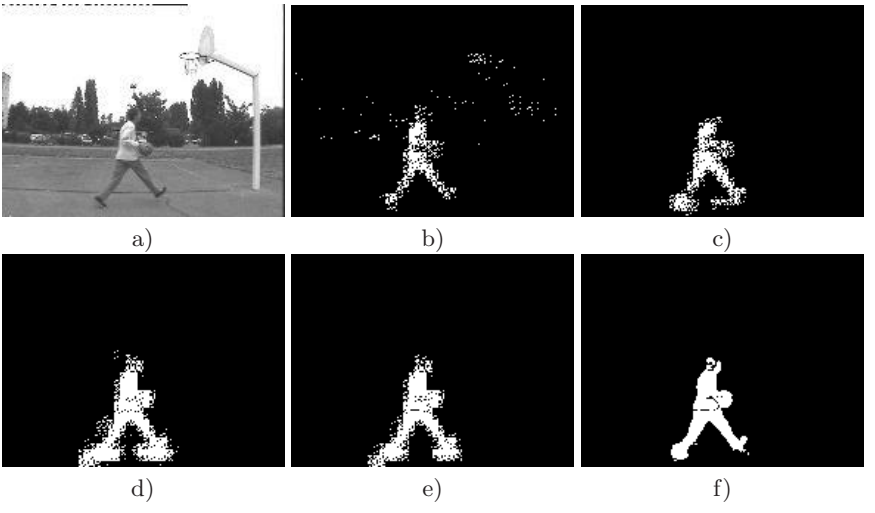


Fig. 3. Detection result for the Basketball sequence. a) Original image, b) Stauffer-Grimson method c) Elgammal et al. method, d) detection with a sequential detection-reconstruction method, without spatial regularization, e) detection using a sequential detection-reconstruction method with regularization, f) detection by our mixed-state method.

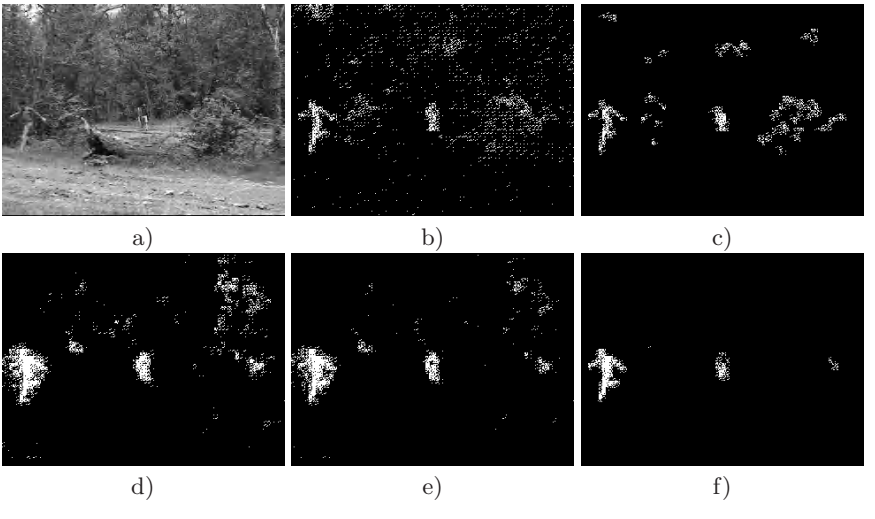


Fig. 4. Detection result for the Forest sequence. a) Original image, b) Stauffer-Grimson method c) Elgammal et al. method, d) detection with a sequential detection-reconstruction method, without spatial regularization, e) detection using a sequential detection-reconstruction method with regularization, f) detection by our mixed-state method.

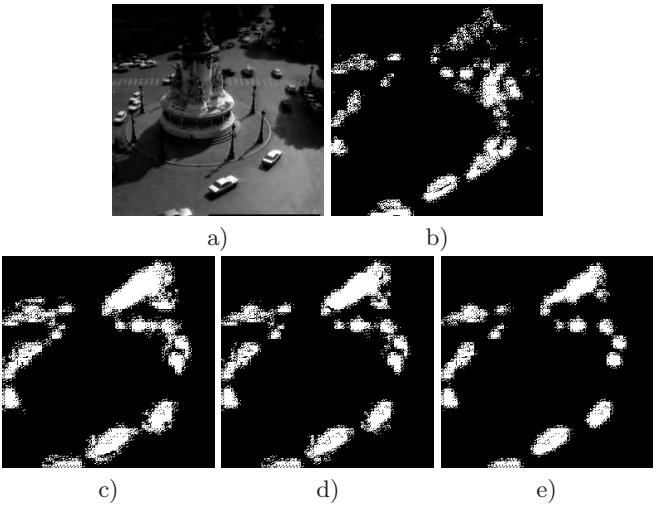


Fig. 5. Detection result for the Traffic Circle sequence. a) Original image, b) Stauffer-Grimson method, c) detection with a sequential detection-reconstruction method, without spatial regularization, d) detection using a sequential detection-reconstruction method with regularization, e) detection by our mixed-state method. The method by Elgammal et al. did not generate a valid result due to the lack of training samples.

the regions occupied by the moving objects are obtained compactly. Note how most of the cars are indeed detected as uniformly connected regions.

5.1 Video Sequence Inpainting

The proposed algorithm generates estimates of the background image, not a model of it, viewed as a problem of reconstruction. The approach uses all the information about the background across time to build a complete image. The importance of this reconstruction not only has implications in the problem of motion detection, but also solves the problem of video sequence inpainting. In this case, moving objects can be removed from the scene as shown in Fig. 6. Moreover, the reconstruction implies smoothing of the background image, over homogeneous intensity regions, filtering out the observation noise, but preserving the edges. In the third row of Fig. 6 we display a small region for each sample, in order to more clearly observe the effect of the background reconstruction. In Fig. 6a), the basketball court is smoothed, and the lines are well preserved. In the Forest sequence 6b), we see how the algorithm preserves the texture of the trees and does not blur the intensity borders. In c), d) and e), the cars are correctly removed even in a complex situation where the background partially occludes the moving object, as in e), and the image noise is reduced as well.

Fig. 6. Top row: original sequences. Center row: background images estimated with our method. Bottom row: a close-up over a small region of the original (left) and reconstructed (right) images. The spatio-temporal reconstruction of the background is achieved jointly with motion detection, resulting in virtually removing the moving objects from the scene. The reference image is also filtered over homogeneous intensity regions in order to reduce noise, but preserving borders.

6 Conclusions

In this paper, we have presented a new approach for addressing a complex problem as simultaneous motion detection and background reconstruction. The interaction between the two tasks was exploited, through a joint decision-estimation formulation, which reduces the problem to a unified step. This improves the regularization of the detection map w.r.t. existing background subtraction methods and against similar but sequential (non-simultaneous) strategies, resulting in more compact and well-defined detected regions. Another original contribution is the introduction of a *conditional* mixed-state random field that allows the integration of motion observations in the scheme.

The implications of considering these types of mixed-state models are enormous in computer vision, where high-level information, represented by abstract labels, can be introduced in an optimal way. Future applications include introduction of symbolic states for: borders (e.g. estimating discontinuous optical flow fields), detection of regions of interest (defined abstractly) or structural change detection (e.g., in remote sensing).

References

1. Kumar, S., Hebert, M.: Discriminative random fields. Int. J. Comput. Vision 68(2), 179–201 (2006)
2. Bouthemy, P., Hardouin, C., Piriou, G., Yao, J.F.: Mixed-state auto-models and motion texture modeling. Journal of Mathematical Imaging and Vision 25(3), 387–402 (2006)

3. Crivelli, T., Cernuschi-Frias, B., Bouthemy, P., Yao, J.F.: Mixed-state markov random fields for motion texture modeling and segmentation. In: Proc. IEEE Int. Conf. on Image Processing (ICIP 2006), Atlanta, USA, pp. 1857–1860 (2006)
4. Salzenstein, F., Pieczynski, W.: Parameter estimation in hidden fuzzy markov random fields and image segmentation. Graph. Models Image Process 59(4), 205–220 (1997)
5. Salzenstein, F., Collet, C.: Fuzzy markov random fields versus chains for multispectral image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 28(11), 1753–1767 (2006)
6. Cernuschi-Frias, B.: Mixed states markov random fields with symbolic labels and multidimensional real values. Technical Report 6255, INRIA (July 2007)
7. Hardouin, C., Yao, J.F.: Spatial modelling for mixed-state observations. Electronic Journal of Statistics (2008)
8. Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P.: Pfinder: real-time tracking of the human body. IEEE Trans. Pattern Anal. Mach. Intell. 19(7), 780–785 (1997)
9. Stauffer, C., Grimson, W.: Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Mach. Intell. 22(8), 747–757 (2000)
10. Elgammal, A.M., Harwood, D., Davis, L.S.: Non-parametric model for background subtraction. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 751–767. Springer, Heidelberg (2000)
11. Parag, T., Elgammal, A., Mittal, A.: A framework for feature selection for background subtraction. In: CVPR 2006: Proc. of the 2006 IEEE Conf. on Computer Vision and Pattern Recognition, Washington, DC, USA, pp. 1916–1923 (2006)
12. Mittal, A., Paragios, N.: Motion-based background subtraction using adaptive kernel density estimation. In: CVPR 2004: Proc. of the 2004 IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2, pp.II–302–II–309 (2004)
13. Sheikh, Y., Shah, M.: Bayesian modeling of dynamic scenes for object detection. IEEE Trans. Pattern Anal. Mach. Intell. 27(11), 1778–1792 (2005)
14. Monnet, A., Mittal, A., Paragios, N., Ramesh, V.: Background modeling and subtraction of dynamic scenes. In: Proc. of the Ninth IEEE Int. Conf. on Computer Vision, vol. 2, pp. 1305–1312 (2003)
15. Sun, J., Zhang, W., Tang, X., Shum, H.Y.: Background cut. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 628–641. Springer, Heidelberg (2006)
16. Bugeau, A., Pérez, P.: Detection and segmentation of moving objects in highly dynamic scenes. In: CVPR 2007: Proc. of the 2007 IEEE Conf. on Computer Vision and Pattern Recognition, Minneapolis, MI (2007)
17. Veit, T., Cao, F., Bouthemy, P.: An a contrario decision framework for region-based motion detection. International Journal on Computer Vision 68(2), 163–178 (2006)

Semidefinite Programming Heuristics for Surface Reconstruction Ambiguities

Ady Ecker, Allan D. Jepson, and Kiriakos N. Kutulakos

Department of Computer Science, University of Toronto

Abstract. We consider the problem of reconstructing a smooth surface under constraints that have discrete ambiguities. These problems arise in areas such as shape from texture, shape from shading, photometric stereo and shape from defocus. While the problem is computationally hard, heuristics based on semidefinite programming may reveal the shape of the surface.

1 Introduction

An important problem in surface reconstruction is the handling of situations in which there are not enough constraints to uniquely determine the surface shape. In these under-constrained situations there are multiple interpretations of the surface that are consistent with the available constraints. The ambiguities can be continuous, such as unknown depth, or discrete, such as in/out reversal. In this paper we deal with constraints that have discrete ambiguities. We show that the problem of integrating a smooth surface under ambiguous constraints can be addressed with semidefinite programming (SDP).

SDP has been applied to a wide range of combinatorial optimization problems. For a general introduction to SDP see [1,2,3]. Recently, SDP-based approximation algorithms have been developed for several computer vision problems, such as image restoration [4,5], segmentation [4,6], graph matching [7,8,9] and finding correspondences in stereo [10]. Zhu and Shi [11] used SDP to solve in/out reversal ambiguities of surface patches in shape from shading.

In this paper we show that a similar mathematical formulation applies to other surface reconstruction problems. Our primary interest is resolving the inherent ambiguities in shape from texture. Similar ambiguities arise in two-light photometric stereo and shape from defocus.

The general approach starts by representing the surface as a spline, i.e. the shape is controlled by a set of continuous variables. Additional discrete variables are used to form the ambiguous constraints. A quadratic cost function measuring surface smoothness and constraint satisfaction is defined. The continuous variables are eliminated, leading to a quadratic cost function in the discrete variables only. An SDP relaxation embeds the discrete variables in a continuous high dimensional space. Finally, a rounding step sets the discrete variables and proposes a 3D shape.

The problems we deal with are larger than those considered by Zhu and Shi, and standard Goemans-Williamson random hyperplane rounding technique [12] will usually produce sub-optimal results. We describe several heuristics that can improve the solution considerably.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 127–140, 2008.

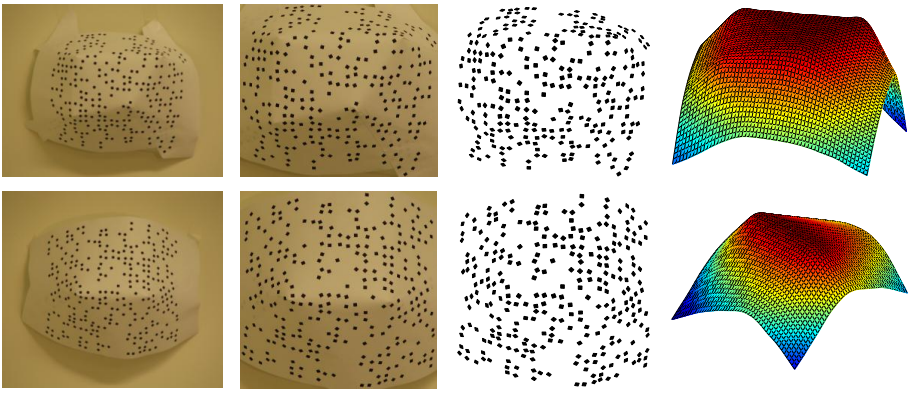


Fig. 1. Left to right: surfaces textured with squares, input images, parallelograms extracted from the images and the computed surfaces

2 Problem Formulation

We show below that several surface reconstruction problems can be written in the form

$$\underset{\boldsymbol{v},\boldsymbol{d}}{\operatorname{argmin}} \|\boldsymbol{A}\boldsymbol{v} - \boldsymbol{B}\boldsymbol{d}\|^2 \ , \tag{1}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are matrices, $\boldsymbol{d} \in \{-1, 1\}^n$ is a vector of discrete decision variables, and $\boldsymbol{v} \in \mathbb{R}^m$ is a vector of continuous parameters that controls the surface. We represent the surface as a spline, i.e. a linear combination of basis functions

$$z(x, y) = \sum_{i=1}^{m} \boldsymbol{b}_i(x, y)\boldsymbol{v}_i \ . \tag{2}$$

The specific bases we use are described in the appendix. Controlling the surface by a relatively low dimensional vector of parameters $\boldsymbol{v}$ reduces the computational load and prevents over-fitting noisy constraints. We discuss several specific problems of this form next. Algorithms for solving (1) are discussed in Sect. 3.

2.1 Shape from Two-Fold Ambiguous Normals

Traditional shape from texture deals with estimating surface normals from the distortion of texture under projection. Under orthographic projection, normal estimates usually have a two-fold tilt ambiguity, because the projections of local planar patches with normals $(p, q, 1)$ and $(-p, -q, 1)$ are identical. We address the problem of estimating the shape of a surface given a large number of ambiguous normal estimates, as demonstrated in Fig. 1.

For sparse texture the problem is under-constrained, since the surface can be integrated for any choices of the normals. In addition, under orthographic projection there

is one global in/out reversal ambiguity, and a continuous ambiguity in absolute depth. However, if we make the assumption that the surface is smooth, we can identify the more probable shapes of the surface. In related work, Forsyth [13,14] proposed alternating between optimizing surface smoothness and selecting the normals. We show in Sect. 3 that by using a quadratic smoothness term the problem can be converted into an entirely discrete optimization problem.

Denote the partial derivatives of the surface by $p = \frac{dz}{dx}$, $q = \frac{dz}{dy}$. The texture observed at a specific image point, say (x_i, y_i), provides two choices for the surface derivatives, namely $(p_i, q_i)\boldsymbol{d}_i$, with $\boldsymbol{d}_i = \pm 1$. By differentiating (2) we can express the derivatives of the spline surface in terms of the vector $\boldsymbol{v}$. This provides two known row vectors $\boldsymbol{a}_{p_i}$ and $\boldsymbol{a}_{q_i}$ such that

$$(0, \ldots, 0, p_i, 0, \ldots, 0)\boldsymbol{d} = \boldsymbol{a}_{p_i} \cdot \boldsymbol{v} \quad , \quad (0, \ldots, 0, q_i, 0, \ldots, 0)\boldsymbol{d} = \boldsymbol{a}_{q_i} \cdot \boldsymbol{v} \ , \tag{3}$$

where $\boldsymbol{d}$ is a n-vector formed from the sign bits $\boldsymbol{d}_i$.

To regularize the surface we use a quadratic smoothness term. The smoothness term can be expressed in terms of the spline parameters using a matrix $\boldsymbol{E}$ such that $\|\boldsymbol{E}\boldsymbol{v}\|^2$ is the smoothness energy (see the appendix for more details). The smoothness energy is weighted with a regularization parameter λ that balances between the smoothness energy and the constraint error. Together, these terms can be written in the form of (1),

$$\underset{\boldsymbol{v},\boldsymbol{d}}{\operatorname{argmin}} \left\| \underbrace{\begin{bmatrix} \sqrt{\lambda}\boldsymbol{E} \\ \boldsymbol{A}' \end{bmatrix}}_{\boldsymbol{A}} \boldsymbol{v} - \underbrace{\begin{bmatrix} 0 \\ \boldsymbol{B}' \end{bmatrix}}_{\boldsymbol{B}} \boldsymbol{d} \right\|^2 . \tag{4}$$

Here $\boldsymbol{A}'$ and $\boldsymbol{B}'$ are matrices formed from the constraints in (3), with each constraint in (3) appearing as a single row in $\boldsymbol{A}'\boldsymbol{v} = \boldsymbol{B}'\boldsymbol{d}$.

2.2 Two-Light Photometric Stereo

In standard photometric stereo of a Lambertian surface, at least three light sources at known positions are used to determine the surface normals. For a given light source, the image brightness at a point constrains the corresponding surface normal to a circle on the unit sphere. Two light sources may limit the normal to two possibilities, which are the intersections of the two circles. The third light disambiguates the normal. In special cases, two lights are sufficient. This occurs when the two circles on the unit sphere touch at a point, or when one of the intersection points of the two circles is on the occluded half-hemisphere. Onn and Bruckstein [15] studied photometric stereo of Lambertian surfaces using two lights. Their method uses the points that are uniquely determined by two lights to divide the image into regions. Inside each region integrability is used to choose between the two possibilities of the normal. However, detecting the boundaries of these regions on a discrete grid is susceptible to errors, especially when the surface has discontinuous derivatives.

Our formulation for two-light photometric stereo avoids region detection and adds a surface smoothness prior. We assume knowledge of points (x, y) where we have two choices for the surface derivatives, (p_1, q_1) or (p_2, q_2). Onn and Bruckstein [15] derived the formulas for the possible derivatives in the Lambertian case. For other shading

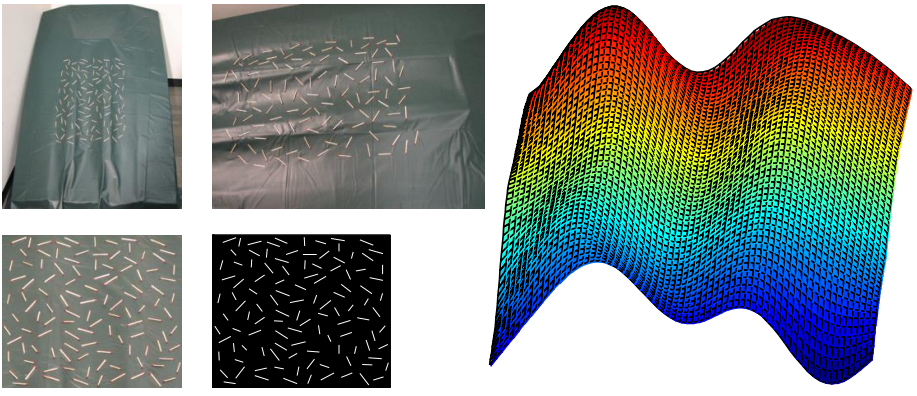


Fig. 2. Top row: setup of 112 matchsticks on a smooth surface. Bottom row: the input image (left) and extracted segments (right). Right: computed surface using 50 basis functions.

models these choices may be determined experimentally. However, more complicated shading models might require more images, since the corresponding two curves on the unit sphere might intersect more than twice. We don't deal with these cases here.

The two choices for the surface derivatives at point (x, y) can be expressed as functions of a sign bit $\boldsymbol{d}_{xy} = \pm 1$

$$\begin{aligned} \begin{pmatrix} p \\ q \end{pmatrix} &= \begin{pmatrix} p_{sum} \\ q_{sum} \end{pmatrix} + \begin{pmatrix} p_{diff} \\ q_{diff} \end{pmatrix} \boldsymbol{d}_{xy} \quad , \\ \begin{pmatrix} p_{sum} \\ q_{sum} \end{pmatrix} &= \frac{1}{2} \begin{pmatrix} p_1 + p_2 \\ q_1 + q_2 \end{pmatrix} \quad , \quad \begin{pmatrix} p_{diff} \\ q_{diff} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} p_1 - p_2 \\ q_1 - q_2 \end{pmatrix} \quad . \end{aligned} \tag{5}$$

As in the previous subsection, the spline derivatives can be written as $p = \boldsymbol{a}_p \cdot \boldsymbol{v}$, $q = \boldsymbol{a}_q \cdot \boldsymbol{v}$. Collecting the equations for all points and adding the smoothness term we arrive at

$$\underset{\boldsymbol{v},\boldsymbol{d}'}{\operatorname{argmin}} \left\| \begin{bmatrix} \sqrt{\lambda}\boldsymbol{E} \\ \boldsymbol{A}' \end{bmatrix} \boldsymbol{v} - \begin{bmatrix} 0 \\ \boldsymbol{B}' \end{bmatrix} \boldsymbol{d}' - \begin{pmatrix} 0 \\ \boldsymbol{b}' \end{pmatrix} \right\|^2 = \underset{\boldsymbol{v},\boldsymbol{d}'}{\operatorname{argmin}} \left\| \begin{bmatrix} \sqrt{\lambda}\boldsymbol{E} \\ \boldsymbol{A}' \end{bmatrix} \boldsymbol{v} - \begin{bmatrix} 0 & 0 \\ \boldsymbol{B}' & \boldsymbol{b}' \end{bmatrix} \begin{pmatrix} \boldsymbol{d}' \\ 1 \end{pmatrix} \right\|^2 . \tag{6}$$

Here the nonzero entries of the matrix $\boldsymbol{B}'$ are made of p_{diff}, q_{diff}, and the vector $\boldsymbol{b}'$ is made of p_{sum}, q_{sum} according to (5). A standard transformation to bring (6) to the form of (1) is to solve for $\boldsymbol{d} = \begin{pmatrix} \boldsymbol{d}' \\ 1 \end{pmatrix}$. Note that the cost of a pair $(\boldsymbol{v}, \boldsymbol{d})$ in (1) equals the cost of $(-\boldsymbol{v}, -\boldsymbol{d})$. Thus, if after solving (6) the last coordinate of $\boldsymbol{d}$ is -1, we need to negate the solution.

2.3 Segments of Known Length

Assume a collection of segments of known 3D length (or pairs of features with known 3D distances) is detected on a smooth surface, as shown in Fig. 2. Given an orthographic

view, each segment can have a front/back reversal. Similar problems were considered by Naito and Rosenfeld [16] and Koenderink and van Doorn [17].

The depth difference of the segment's endpoints is constrained by

$$z_i - z_j = \boldsymbol{d}_{ij}\sqrt{l^2 - r_{ij}^2} = \Delta_{ij}\boldsymbol{d}_{ij} \ , \tag{7}$$

where $\boldsymbol{d}_{ij} = \pm 1$, l is the 3D length of the segment and r_{ij} is the observed length in the image. l, r_{ij} (and hence Δ_{ij}) are assumed to be known. By (2), the depth z_i at a point (x_i, y_i) is a linear combination of the spline bases at that point. That is, z_i can be expressed as $\boldsymbol{a}_i \cdot \boldsymbol{v}$, where $\boldsymbol{a}_i$ is a known row vector, and similarly for z_j. Each depth constraint can be written as

$$(\boldsymbol{a}_i - \boldsymbol{a}_j)\boldsymbol{v} = (0, \ldots, 0, \Delta_{ij}, 0, \ldots, 0)\boldsymbol{d} \ . \tag{8}$$

Collecting these equations over all constraints and adding the smoothness term can be written as in (4).

2.4 Shape from Defocus

In shape from defocus, depth is estimated by measuring blur level differences between multiple images taken with different focus and aperture camera settings. Here we consider a simple case involving just two images, both taken with the same focus setting. One image is obtained using a small aperture and is assumed to be sharp. The second image is taken with a large aperture and exhibits more blurring. The blur of a local region is measured as the standard deviation σ of a Gaussian that needs to be convolved with the sharp image to match the blurred image. The estimated $\sigma > 0$ for an image patch corresponds to a two-fold ambiguity for the depth, with one depth in front of the in-focus plane and the other behind. The formulas relating σ and the camera parameters to the two possible depths were developed by Pentland [18]. In Sect. 4 we experiment with a first-order simplified model that assumes the depth is proportional to $\pm\sigma$. This model falls naturally into the form (4), where now the matrix $\boldsymbol{A}'$ contains the depths of the spline bases vectors, and the matrix $\boldsymbol{B}'$ the estimated σ for a collection of points. Alternatively, Pentland's model could be expressed in the same general form using sums and differences as in (5) and (6).

Note that this example is meant only to demonstrate the formulation, as the practical use of this approach is rather limited. For points far from the in-focus plane the blur in the second image may be too heavy for us to reliably resolve σ. Moreover, for points in the second image which are nearly at the focused depth, it is again difficult to get an unbiased estimate of σ. As a consequence, useful information from the aperture change is only available over a narrow range of depths. In practice it is simpler to place the object on one side of the in-focus plane or take more images with different focus settings.

3 SDP Rounding Heuristics

We now turn into solving problems of the form (1). For any vector $\boldsymbol{d}$, the optimal $\boldsymbol{v}$ is

$$\boldsymbol{v} = \boldsymbol{A}^{+}\boldsymbol{B}\boldsymbol{d} \ , \tag{9}$$

where $\boldsymbol{A}^+$ is the pseudo-inverse of $\boldsymbol{A}$. Plugging $\boldsymbol{v}$ back into equation (1) we get $\|(\boldsymbol{A}\boldsymbol{A}^+\boldsymbol{B}-\boldsymbol{B})\boldsymbol{d}\|^2 = \boldsymbol{C}\bullet\boldsymbol{X}$, where $\boldsymbol{C} = (\boldsymbol{A}\boldsymbol{A}^+\boldsymbol{B}-\boldsymbol{B})^t(\boldsymbol{A}\boldsymbol{A}^+\boldsymbol{B}-\boldsymbol{B}) = \boldsymbol{B}^t(\boldsymbol{I}-\boldsymbol{A}\boldsymbol{A}^+)\boldsymbol{B}$, $\boldsymbol{X} = \boldsymbol{d}\boldsymbol{d}^t$, and $\bullet$ is the inner product of matrices ($\boldsymbol{C}\bullet\boldsymbol{X} = \sum \boldsymbol{C}_{ij}\boldsymbol{X}_{ij}$). Therefore, the problem is reduced into a combinatorial optimization problem of finding the discrete vector $\boldsymbol{d}\in\{-1,1\}^n$ which minimizes $\boldsymbol{C}\bullet(\boldsymbol{d}\boldsymbol{d}^t)$. Once $\boldsymbol{d}$ is found, $\boldsymbol{v}$ is given by (9) and the 3D shape is given by (2).

Unfortunately the general problem is NP-hard and difficult to approximate [19]. Semidefinite programming is widely used to find approximate solutions for problems of this kind. The standard SDP relaxation requires the matrix $\boldsymbol{X}$ to be symmetric positive semidefinite (instead of rank one) with ones on the main diagonal (since $\boldsymbol{d}_i^2 = 1$), i.e. solving

$$\underset{\boldsymbol{X}}{\operatorname{argmin}}\ \boldsymbol{C}\bullet\boldsymbol{X} \quad \text{s.t.}\ \ \boldsymbol{X}_{ii} = 1\ ,\ \ \boldsymbol{X}\succeq 0\ . \tag{10}$$

Since this problem is convex, the relaxation can be solved in polynomial time using an SDP solver. A discrete vector $\boldsymbol{d}$ is obtained from the continuous solution matrix $\boldsymbol{X}$ in a rounding phase. The Goemans-Williamson random hyperplane rounding scheme [12] uses the Cholesky factorization of the matrix $\boldsymbol{X}$, $\boldsymbol{X} = \boldsymbol{R}\boldsymbol{R}^t$. Let $\boldsymbol{u}_i$ denote the i-th row of $\boldsymbol{R}$. Since $\boldsymbol{X}_{ii} = \boldsymbol{u}_i\cdot\boldsymbol{u}_i^t = 1$, the rows of $\boldsymbol{R}$ can be viewed as an embedding of the decision variables into the unit sphere in $\mathbb{R}^n$ (this embedding is not unique). Rounding is done by picking a random hyperplane with normal $\boldsymbol{N}$ and setting $\boldsymbol{d}_i = sign(\boldsymbol{u}_i\cdot\boldsymbol{N})$.

For matrices $\boldsymbol{C}$ with nonnegative entries arising from the max-cut problem, this scheme provides a strong, provable expected approximation ratio of at least 0.878. However, this result does not apply directly to our problem for several reasons. First, their analysis is for the cost of the resulting cut, not the quadratic cost function itself. Secondly, our matrices may have negative elements. Thirdly, we are interested in the shape of the surface, not the number of correctly classified sign bits. A small number of misclassifications may have large influence on the shape or may not be visible at all. Note that it is possible to have different solutions with very different shapes but similar objective values. It is also possible that the correct surface is not the minimal solution (e.g. when the correct surface is not smooth, or when there are insufficient or noisy constraints). This depends on the instance of the problem.

In our setting, random hyperplane rounding typically requires a huge number of iterations to produce high quality results. We apply a series of heuristics for improvement:

1. Instead of picking plane normals from a uniform distribution on the sphere, we use the principle singular vectors of $\boldsymbol{R}$ (that correspond to largest singular values). For example, if we had to choose a single plane, a good choice for the normal would be the principle singular vector. Such "inertial" splitting methods have been previously used for other embeddings [20,21], but to our best knowledge not for the SDP embedding. To widen the choices of planes, we randomly pick normals as a weighted linear combination of the singular vectors that correspond to the k-largest singular values, i.e.

$$\boldsymbol{N} = \sum s_i\lambda_i\boldsymbol{w}_i\ , \tag{11}$$

where $s_i \sim N(0,1)$ and $\boldsymbol{w}_i$ is the singular vector corresponding to the singular value λ_i. Finding these singular vectors can be done efficiently by power-iteration methods.

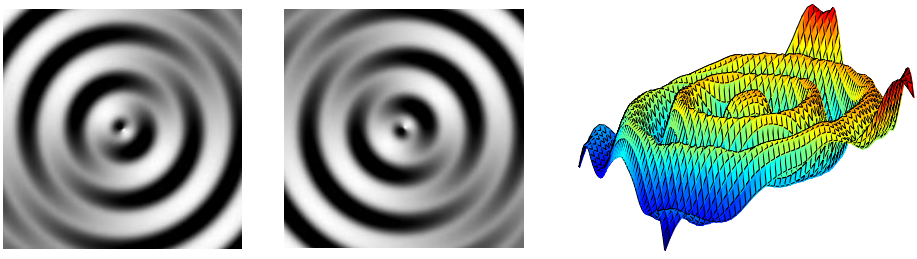


Fig. 3. Surface computed from a synthetic photometric stereo pair

2. Instead of making the decision based on a single normal, we randomly select a pair of normals $\boldsymbol{N}_1, \boldsymbol{N}_2$ according to (11), and perform a circular sweep for normals in the plane spanned by $\boldsymbol{N}_1, \boldsymbol{N}_2$. To do this, we project the points embedded in $\mathbb{R}^n$ on this plane (the first plane we check is the one spanned by the two principle singular vectors). Then we perform a circular sweep in this plane, as described in [22]. Basically, the sweep rotates a line through the origin that separates the points into two groups, and picks the partition with the lowest cost. We noticed that these angular sweeps can be made more efficient by careful bookkeeping similar to the Kernighan-Lin (K-L) algorithm [23]. Note that the cost of splitting n points in $\mathbb{R}^n$ based on a single normal is $O(n^2)$. However, scanning a series of n normals, where at each transition a single point moves to the other side of the sweep line, can also be carried out in $O(n^2)$.
3. The k-best results from the circular sweep phase are refined with the K-L algorithm [23,24,25,26]. This is a local search procedure that will clean up a small number of misplaced vertices. We terminate this algorithm early if no progress is made in 50 consecutive iterations [25]. The lowest cost solution among these trials is returned.

4 Two-Fold Ambiguity Results

Figure 1 demonstrates reconstruction from ambiguous normals. To simplify texture extraction we used square texture elements (see Sect. 5 for derivation of normals from parallelograms). The SDP solver we used is DSDP [27]. In Fig. 3 we computed a surface from a pair of synthetic images of a Lambertian surface using two-light photometric stereo. The two possibilities for the surface normal are computed on a 29×29 grid. Points having only a single possible normal and points in attached shadow were removed. The system had 425 discrete variables. Our program made the correct decision at each of these points. However, the average deviation from the true surface is 19% (note that two corners are in shadow).

Results for segments of known length are shown in Figs. 2 and 4. In Fig. 2, the image was taken from a distance of about 10m with a zoom lens to approximate orthographic projection. As suggested by Naito and Rosenfeld [16], the 3D length of the segments is estimated as the maximum over the 2D lengths of all segments in the image. Figure 4 shows 1521 randomly oriented line segments tangent to a synthetic surface (top and

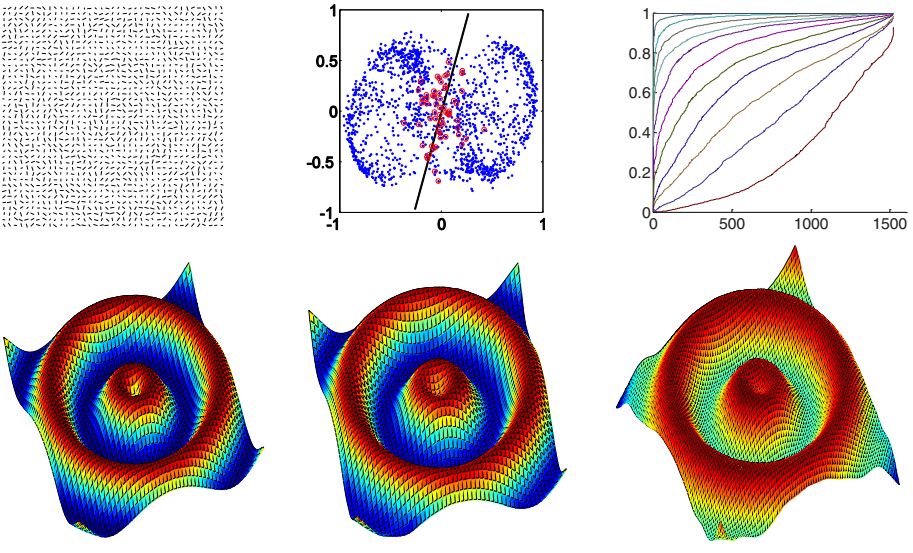


Fig. 4. Top left: segments of known and equal length on synthetic surface. Top center: projection of the SDP embedding on the subspace of the first two principle components. Misclassified points with respect to the ground truth are circled. Top right: plot of the squared-length of the projections of the embedded points on the first principle subspaces (see text). Bottom left: original surface. Bottom middle: output of the program. Bottom right: output using random hyperplane rounding.

bottom left). The top-center plot shows the projection of the SDP embedding on the subspace of the first two principle components. The splitting line is the lowest energy circular cut for this projection. To check the appropriateness of the projection on a low-dimensional subspace we projected the SDP embedding on the subspaces spanned by the first $1, 2, 3, \ldots$ principle vectors. The top-right plot is the squared-length of these projections, where the 1521 values are sorted. The lowest curve is the distribution of magnitudes of the projection on the subspace of the first principle vector; the second curve is the distribution of magnitudes for the projection on the subspace of the first two principle vectors, etc. It is evident that the points were embedded near a low dimensional subspace, and this is used for more efficient rounding. The solution (bottom-center) was computed using 300 spline bases, 1000 circular sweeps on random planes, and K-L runs on the best 100 vectors. The program made 15 wrong decisions, and the average height deviation from the truth surface is 1%. In comparison, a run of the Goemans-Williamson random hyperplane rounding (bottom right) with 10^4 trials produced 82 misclassifications, with 2.6% average height deviation. While these numbers depend very much on the particular instance, this example shows that the SDP approach to ambiguities can deal with much larger instances than demonstrated by Forsyth [13].

Figure 5 demonstrates shape reconstruction from defocus using two images taken with a narrow and a wide aperture. The camera is focused at the middle of the shape. The blur level is estimated over a grid of 15×60 points. To estimate the blur level, we convolved a window of the narrow-aperture image around each grid point with Gaussians of various sizes and pick the size that best matches the blurred image. To resist

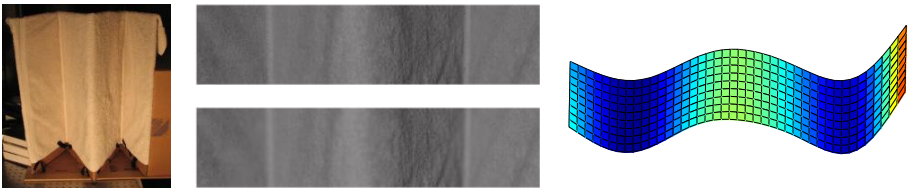


Fig. 5. Shape reconstruction from defocus. Left: the imaged surface. Middle: the sharp (narrow aperture) and blurred (wide aperture) input images. Right: the computed surface.

noise, we pick the median of 9 windows near a grid point. Points where the estimated blur level is very low or very high were removed since, for these points, the relation between the estimated defocus and the depth is inaccurate.

5 Four-Fold Ambiguities

The previous sections looked at problems where each decision had two options. In this section we demonstrate an extension to discrete ambiguities with four options. As a model, we look at a shape from texture problem. Suppose a collection of similar triangles are scattered on a smooth surface viewed orthographically. All triangles are scaled versions of a known triangle. In addition, we assume that one edge on each triangle can be identified. For instance, if the texture elements are rectangles, as in Fig. 6, we can identify the diagonal. This information leads to four-fold ambiguity since there are two ways to match the image segments with the edges of the known triangle.

Algebraically, let (x_i, y_i, z_i), $i = 1, 2, 3$, be the three vertices of a triangle in the image. Denote $dx_{ij} = x_i - x_j$, $dy_{ij} = y_i - y_j$, $dz_{ij} = z_i - z_j = p \cdot dx_{ij} + q \cdot dy_{ij}$, where p, q are the slopes of the triangle's plane. Since the triangle is similar to the model triangle, the 3D length ratios are known

$$r_1 = \frac{dx_{12}^2 + dy_{12}^2 + dz_{12}^2}{dx_{13}^2 + dy_{13}^2 + dz_{13}^2} \quad , \quad r_2 = \frac{dx_{23}^2 + dy_{23}^2 + dz_{23}^2}{dx_{13}^2 + dy_{13}^2 + dz_{13}^2} \ . \tag{12}$$

This leads to two quadratic equations in p, q. Simple manipulations lead to a quadratic equation in q^2 that can be solved for the positive root (details are omitted). Switching between r_1,r_2 gives another solution. In the general case, the four solutions are of the form $\pm(p_1, q_1)$ and $\pm(p_2, q_2)$.

In the SDP literature, the max-k-cut problem was studied by Frieze and Jerrum [28] and de Klerk, Pasechnik, and Warners [29]. While an ideal encoding requires two bits to encode four possibilities, their encoding uses four bits: a single indicator bit set to 1 and the rest 0. Since the matrix $\boldsymbol{X}$ has $O(n^2)$ entries, redundant encoding makes the SDP problem 4 times larger, which is a significant factor for current SDP solvers. There is a natural encoding of this problem with two sign bits for each constraint using average and offset vectors similar to the sums and difference vectors of Sect. 2.2. However, we found that if the two bits are completely independent, the rounding phase becomes more difficult and results get worse. Instead, our encoding uses two variables d_1, d_2 for each

triangle that ideally would take the values $-1, 0, 1$. We add the constraint $d_1 \cdot d_2 = 0$ so that only one decision variable is active at a time

$$p = d_1 p_1 + d_2 p_2 \quad , \quad q = d_1 q_1 + d_2 q_2 \quad , \quad d_1 \cdot d_2 = 0 \ . \tag{13}$$

By expressing p, q for each triangle using the spline parameters and adding the smoothness term, we arrive at a system of the form (1) in $2n$ discrete variables. The SDP relaxation is modified to

$$\underset{\boldsymbol{X}}{\operatorname{argmin}} \ \boldsymbol{C} \bullet \boldsymbol{X} \quad \text{s.t.} \ \boldsymbol{X}_{2i-1,2i-1} + \boldsymbol{X}_{2i,2i} = 1 \ , \ \boldsymbol{X}_{2i-1,2i} = 0 \ , \ \boldsymbol{X} \succeq 0 \ . \tag{14}$$

After solving the SDP problem (14) for $\boldsymbol{X}$, Cholesky factorization $\boldsymbol{X} = \boldsymbol{R}\boldsymbol{R}^t$ embeds the decision variables into a sphere in $\mathbb{R}^{2n}$. Let $\boldsymbol{u}_i$ denote the i-th row of $\boldsymbol{R}$, so that $\boldsymbol{X}_{ij} = \boldsymbol{u}_i \cdot \boldsymbol{u}_j^t$. For each decision there are two orthogonal vectors, $\boldsymbol{u}_{2i-1}$,$\boldsymbol{u}_{2i}$, such that $\|\boldsymbol{u}_{2i-1}\|^2 + \|\boldsymbol{u}_{2i}\|^2 = 1$. The rounding phase has to decide which of the two vectors is active, and round the active variable to either 1 or -1. In the ideal case, the vectors associated with the inactive variables would concentrate near the origin. While concentration can be observed, deciding which variable is active by picking the longer vector of each pair is not powerful enough. We execute the following heuristics:

1. A column of the matrix $\boldsymbol{X}$ provides an indication about the orthogonality between a row vector $\boldsymbol{u}_i$ of $\boldsymbol{R}$ to the other rows. We multiply $\boldsymbol{u}_i$ by α_i, the magnitude of the i-th column of $\boldsymbol{X}$. This has the effect of moving vectors orthogonal to the rest (in particular close to 0) towards the origin. In addition, the largest singular vectors of the modified vectors become less affected by the inactive variables.
2. The vectors $\alpha_i \boldsymbol{u}_i$ are projected on a plane as in (11), and a line is swept circularly. For each pair, the point with the largest projection on the sweep line is considered active, and the sign of the projection is the rounded value (events in this circular sweep occur at angles where the projections of the points on the sweep line are equal in absolute value). We repeat this step for different planes and store the best circular cut for each plane.
3. For every variable, we estimate the probability p_i it is inactive as the percentage of best cuts where it was inactive. Clearly, $p_{2i-1} + p_{2i} = 1$. We modify the diagonal of the matrix $\boldsymbol{C}$ to reflect this knowledge by setting $\boldsymbol{C}_{ii} = \boldsymbol{C}_{ii} + \mu p_i$ (μ is a tuning parameter). The SDP is solved a second time with the modified matrix $\boldsymbol{C}$. Due to the constraint $\boldsymbol{X}_{2i-1,2i-1} + \boldsymbol{X}_{2i,2i} = 1$, vectors which are believed to be inactive are pushed towards the origin.
4. We repeat steps 1,2 with the modified $\boldsymbol{C}$, keep the best solutions and run the K-L algorithm as a final step. The modification of the K-L algorithm to four possibilities is straightforward.

The method is demonstrated in Figs. 6 and 7. In Fig. 6, A collection of 415 similar rectangles at random orientations is overlaid on a 3D surface. Our program uses triangles made of the diagonal and two sides of each rectangle. Only the proportions of the model triangle are assumed known. The circular sweep in the plane of the first two principle components makes 148 errors in the first round (out of 830), and after modifying the $\boldsymbol{C}$ matrix, makes 114 errors in the second round. Note that a large number of

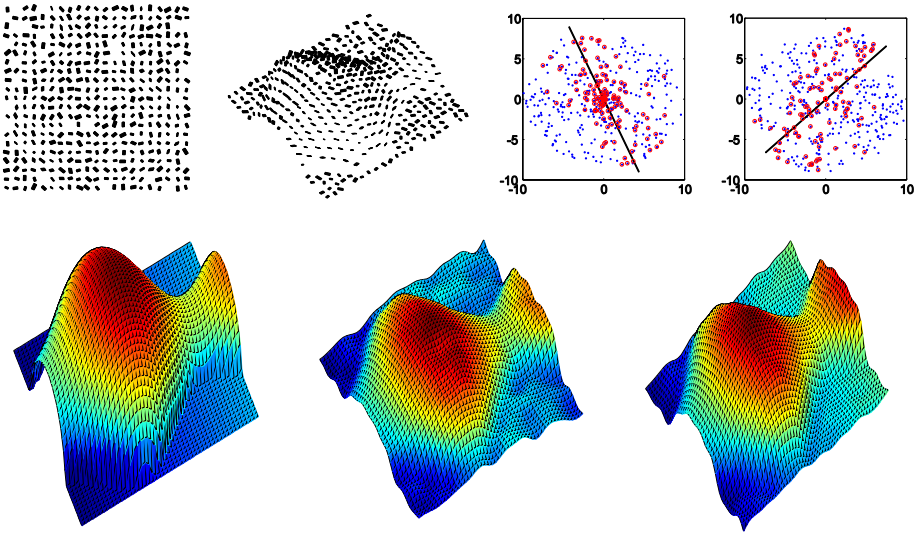


Fig. 6. Reconstruction from similar rectangles. Top left: the input image is made of similar rectangles overlaid on a 3D surface. Top right: the circular cuts of the projections on the plane of the two principle directions for the first and second SDP embeddings. Bottom Left: original surface. Bottom middle: output of the program. Bottom right: spline surface using the ground truth decision vector.

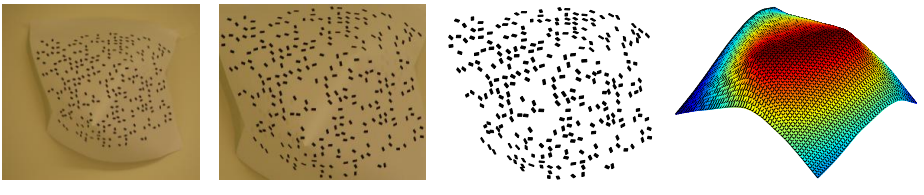


Fig. 7. Left to right: surface textured with rectangles in ratio 1:2, input image, parallelograms extracted from the image and the computed surface

inactive variables are concentrated too close to the origin than could be distinguished in this figure. The final computed surface (bottom middle) makes 80 errors. The average height difference between the original and computed surfaces is 5%. Wiggles in the computed surface arise because only 300 basis are used. Similar wiggles occur with a spline that uses the ground truth decision vector (bottom right). In this example, the cost function of the computed surface is lower than the cost of the ground truth spline.

6 Conclusions

Several depth cues, such as texture, shading and defocus, are inherently ambiguous at the local level. In this paper we examined integration of discrete constraints arising from these ambiguities with continuous objectives like surface smoothness. Problems

of this form involve both continuous and discrete variables, and can be transformed into an entirely discrete optimization problem, which is computationally hard. For an approximate solution we used SDP relaxation. We improved the rounding phase using a combination of heuristics, namely projection on planes in the subspace of the largest principle components, efficient circular sweep, and the K-L algorithm. These general heuristics were shown to be useful in our setting, and are potentially useful for other SDP applications as well.

Compared to other energy minimization approaches, such as belief propagation or graph cuts, our approach is global. Any discrete decision directly influences the cost of the entire surface. The optimization is not based on local neighborhoods. Using a global approach is sometimes necessary, since knowledge of a label at a point could say little about a neighboring label. Another important property of SDP is that there is no need for initialization. However, if a good starting vector $\boldsymbol{d}$ is available, it can be exploited by performing the rounding phase on a linear combination of $\boldsymbol{X}$ and $\boldsymbol{d} \cdot \boldsymbol{d}^t$ [30].

Our focus has been on developing a general framework for solving problems of the form (1), involving ambiguous discrete constraints. In practice, after binary decisions are made, the surface can be re-integrated using more robust integration methods. While the use of smoothness was demonstrated to resolve certain shapes, for many surfaces smoothness alone is insufficient and additional unambiguous constraints are required. A natural extension to the approach presented here would be to replace the spline with a shape basis, e.g. for particular shapes such as faces [31]. The general form of (1) allows for many other variations, such as adding linear constraints (e.g. specifying depths or normals at specific points [32]), or using shading information to disambiguate some normals [33].

References

1. Helmberg, C.: Semidefinite programming for combinatorial optimization. Technical Report ZIB-Report ZR-00-34, TU Berlin (2000)
2. Laurent, M., Rendl, F.: Semidefinite programming and integer programming. In: Handbook on Discrete Optimization, pp. 393–514. Elsevier, Amsterdam (2005)
3. Todd, M.J.: Semidefinite optimization. Acta Numerica 10, 515–560 (2001)
4. Keuchel, J., Schnorr, C., Schellewald, C., Cremers, D.: Binary partitioning, perceptual grouping, and restoration with semidefinite programming. PAMI 25(11), 1364–1379 (2003)
5. Keuchel, J.: Multiclass image labeling with semidefinite programming. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 454–467. Springer, Heidelberg (2006)
6. Carl Olsson, A.E., Kahl, F.: Solving large scale binary quadratic problems: Spectral methods vs. semidefinite programming. In: CVPR 2007, pp. 1–8 (2007)
7. Bai, X., Yu, H., Hancock, E.: Graph matching using spectral embedding and semidefinite programming. In: BMVC 2004, pp. 297–307 (2004)
8. Yu, H., Hancock, E.R.: Graph seriation using semi-definite programming. In: Brun, L., Vento, M. (eds.) GbRPR 2005. LNCS, vol. 3434, pp. 63–71. Springer, Heidelberg (2005)
9. Schellewald, C., Schnörr, C.: Probabilistic subgraph matching based on convex relaxation. In: Rangarajan, A., Vemuri, B.C., Yuille, A.L. (eds.) EMMCVPR 2005. LNCS, vol. 3757, pp. 171–186. Springer, Heidelberg (2005)

10. Torr, P.: Solving markov random fields using semi definite programming. In: Proc. Ninth International Workshop on Artificial Intelligence and Statistics (2003)
11. Zhu, Q., Shi, J.: Shape from shading: Recognizing the mountains through a global view. In: CVPR 2006, pp. 1839–1846 (2006)
12. Goemans, M.X., Williamson, D.P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. J. ACM 42(6), 1115–1145 (1995)
13. Forsyth, D.: Shape from texture and integrability. In: ICCV 2001, pp. 447–452 (2001)
14. Forsyth, D.: Shape from texture without boundaries. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 225–239. Springer, Heidelberg (2002)
15. Onn, R., Bruckstein, A.: Integrability disambiguates surface recovery in two-image photometric stereo. Int. J. Comput. Vision 5(1), 105–113 (1990)
16. Naito, S., Rosenfeld, A.: Shape from random planar features. CVGIP 42(3), 345–370 (1988)
17. Koenderink, J., van Doorn, A.: Shape from chebyshev nets. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 215–225. Springer, Heidelberg (1998)
18. Pentland, A.P.: A new sense for depth of field. PAMI 9(4), 523–531 (1987)
19. Arora, S., Berger, E., Hazan, E., Kindler, G., Safra, M.: On non-approximability for quadratic programs. In: FOCS 2005, pp. 206–215 (2005)
20. Chan, T.F., Gilbert, J.R., Teng, S.H.: Geometric spectral partitioning. Technical Report Tech. Report CSL-94-15, Xerox PARC (1995)
21. Fjallstrom, P.O.: Algorithms for graph partitioning: A survey. Linkoping Electronic Articles in Computer and Information Science 3(10) (1998)
22. Burer, S., Monteiro, R.D.C., Zhang, Y.: Rank-two relaxation heuristics for max-cut and other binary quadratic programs. SIAM J. on Optimization 12(2), 503–521 (2002)
23. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. The Bell system technical journal 49(1), 291–307 (1970)
24. Fiduccia, C., Mattheyses, R.: A linear-time heuristic for improving network partitions. In: Proc. 19th Design Automation Conference, pp. 175–181 (1982)
25. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. 20(1), 359–392 (1998)
26. Krishnan, K., Mitchell, J.E.: A semidefinite programming based polyhedral cut and price approach for the maxcut problem. Comp. Optim. Appl. 33(1), 51–71 (2006)
27. Benson, S.J., Ye, Y.: Algorithm 875: DSDP5: Software for semidefinite programming. ACM Trans. Math. Software 34(3) (2008)
28. Frieze, A., Jerrum, M.: Improved approximation algorithms for maxk-cut and max bisection. Algorithmica 18(1), 67–81 (1997)
29. de Klerk, E., Pasechnik, D.V., Warners, J.P.: On approximate graph colouring and max-k-cut algorithms based on the theta-function. J. Comb. Optim. 8(3), 267–294 (2004)
30. Rendl, F., Rinaldi, G., Wiegele, A.: Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. Technical report, Alpen-Adria-Universität Klagenfurt, Inst. f. Mathematik (2007)
31. Atick, J.J., Griffin, P.A., Redlich, A.N.: Statistical approach to shape from shading: Reconstruction of three-dimensional face surfaces from single two-dimensional images. Neural Computation 8(6), 1321–1340 (1996)
32. Zhang, L., Dugas-Phocion, G., Samson, J.S., Seitz, S.M.: Single view modeling of free-form scenes. In: CVPR 2001, pp. 990–997 (2001)
33. White, R., Forsyth, D.: Combining cues: Shape from shading and texture. In: CVPR 2006, pp. 1809–1816 (2006)

Appendix

In this section we describe in more detail the spline (2) and smoothness energy used in our implementation. These were chosen mainly for the sake of simplicity, and other splines and energies (e.g. thin-plate spline) could be used. To simplify notation assume the image is square. We used tensor-product spline bases

$$\boldsymbol{b}_{ij}(x,y) = \boldsymbol{b}_i(x)\boldsymbol{b}_j(y) \ , \tag{15}$$

where $\boldsymbol{b}_i, \boldsymbol{b}_j$ are singular vectors associated with small singular values of the matrix

$$\boldsymbol{D} = \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & & \cdots & 1 & -2 & 1 \end{bmatrix} \ . \tag{16}$$

For the smoothness energy we used the sum of squared second derivatives over the image. The energy of a basis function of the form (15), with $\|\boldsymbol{b}_i\| = \|\boldsymbol{b}_j\| = 1$, is

$$e_{ij}^2 = \|\boldsymbol{D}\boldsymbol{b}_i\|^2 + \|\boldsymbol{D}\boldsymbol{b}_j\|^2 \approx \iint \left(\frac{d^2\boldsymbol{b}_{ij}}{dx^2}\right)^2 + \left(\frac{d^2\boldsymbol{b}_{ij}}{dy^2}\right)^2 dx\, dy \ . \tag{17}$$

Note that for tensor-product splines we need to integrate only one-dimensional functions. The vectors $\boldsymbol{D}\boldsymbol{b}_i$ are proportional to the left singular vectors of $\boldsymbol{D}$ and hence orthogonal. Since the basis functions $\boldsymbol{b}_i$ are orthogonal, as are $\boldsymbol{D}\boldsymbol{b}_i$, the smoothness of a spline governed by $\boldsymbol{v}$ can be written as $\|\boldsymbol{E}\boldsymbol{v}\|^2$, where $\boldsymbol{E}$ is a diagonal matrix made of the elements e_{ij} for each basis used. The advantage of this approach over Fourier basis is that the basis vectors are not cyclic.

Robust Optimal Pose Estimation

Olof Enqvist and Fredrik Kahl

Lund University, Box 118, 22100 Lund, Sweden

Abstract. We study the problem of estimating the position and orientation of a calibrated camera from an image of a known scene. A common problem in camera pose estimation is the existence of false correspondences between image features and modeled 3D points. Existing techniques such as RANSAC to handle outliers have no guarantee of optimality. In contrast, we work with a natural extension of the L_∞ norm to the outlier case. Using a simple result from classical geometry, we derive necessary conditions for L_∞ optimality and show how to use them in a branch and bound setting to find the optimum and to detect outliers. The algorithm has been evaluated on synthetic as well as real data showing good empirical performance. In addition, for cases with no outliers, we demonstrate shorter execution times than existing optimal algorithms.

1 Introduction

Camera pose estimation is a well studied problem in both computer vision and photogrammetry [1,2] and one of the earliest references on the topic dates back to 1841 [3]. The problem is important on its own as a core problem within the field of multiple view geometry, but it also appears as a subproblem for many other vision applications, like motion segmentation [4], object recognition [5,6,7], and more generally model matching and fitting problems, see [4,5,6,7,8,9,10]. Yet, previous approaches for solving the camera pose problem have not been able to solve the problem in the presence of outliers with a guarantee of global optimality. In this paper, an efficient algorithm is developed for achieving these criteria.

Often the determination of camera pose is divided into two steps. First feature points are extracted from the image and matched to a 3D model of the scene. In the next step, correspondence pairs from the matching procedure are used to estimate the position and orientation of the camera. It is this second step that we will consider in this article for the pinhole camera model.

Fitting problems with outliers are known to be hard optimization problems. We formulate the pose problem as a mixed integer problem and apply branch and bound to find the optimal solution. For such problems it is important to have bounding functions that are fast to compute and give hard constraints on the solution. We derive a bounding function which fulfills both these requirements using a classical result from geometry. This is the key component for the main contribution of the paper: an efficient, robust and optimal solution to the pose problem.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 141–153, 2008.

1.1 Related Work

The minimal number of correspondence pairs necessary to obtain a solution for the camera pose problem is three and it is well-known that there may be up to four solutions in this case, see [11] for an overview. For four points, it can be solved linearly [12] and for six points or more, one can use the standard Direct Linear Transform (DLT) scheme [1], but these approaches optimize an algebraic cost function. Refinement using non-linear optimization techniques like Levenberg-Marquardt is of course possible, but the solution may get trapped in a local minimum. Recent work in multiple view geometry has focused on obtaining global solutions, see [13] for an overview, and this paper is no exception in that sense.

The first globally optimal algorithm for this problem using a geometric error norm was presented in [14]. They also investigate the problem of local minima for the pose problem and show that this is indeed a real problem for small number of correspondences. For large number of correspondence pairs or small noise levels, the risk of getting trapped in a local minimum is small. This is our experience as well. In their work the L_2 norm of the reprojection errors is used, but the algorithm converges rather slowly. In [15], the authors choose to work with the L_∞ norm instead and present a more efficient algorithm that finds the optimum by searching in the space of rotations and solving a series of second order cone programs. The reported execution times for both these algorithms are in the order of several minutes, while our algorithm performs the same task within a few seconds. Perhaps more importantly though is that our algorithm is capable of discarding outliers. If some correspondences are incorrect, then fitting a solution to all data might give a very bad result.

A method for detecting outliers for L_∞ solutions was given in [16] but it only applies to quasiconvex problems. The uncalibrated pose problem - often refered to as camera resectioning - is quasiconvex, but the calibrated pose problem is not. Further, the strategy in [16] for removing outliers is rather crude - all measurements that are in the support set are discarded. Hence, inlier correspondences may also be removed. Possible solutions to this problem was given in [17,18] for outliers, but they are computationally expensive and also restricted to quasiconvex problems. Another well-known approach for estimating camera pose in cases where it is hard to find correct correspondences is to apply RANSAC-type algorithms [19]. Such algorithms offer no type of optimality.

Our work is also related to the rich body of literature on matching problems, see [4,5,6,7,8,9,10]. Many of these algorithms are quite sophisticated and have been an inspiration to our work. However, some do not guarantee any kind of optimality [4,10], while others do [8,9]. Another difference to our work is that simplified camera models like affine approximations are used [6,7,8,9]. Other drawbacks include simplified cost functions which are not based on reprojection errors.

2 Problem Formulation

Given an image from a calibrated camera and a 3D model of the scene, we want to estimate the position C, and orientation $\boldsymbol{R}$, of the camera. If we have more

than three points this is generally an overdetermined problem. Thus we cannot expect an exact solution and need to decide on an error measure. The widely accepted standard is to consider some error norm on the reprojection errors. We will follow this standard as well.

Since the camera is calibrated we can choose to represent the image as a sphere (rather than an image plane) and detected feature points in the image as points on the sphere or unit vectors, x_j. We also have a set of hypothetical correspondences between points in the model and points in the image (X_i, x_i). We will work with two natural extensions of the L_∞ norm to the outlier case.

Problem 1. *For a prescribed threshold $\varepsilon > 0$, find the rotation $\boldsymbol{R}$ and position C that maximizes $|I|$ where I is the set of indices i such that*

$$\angle\left(x_i\,,\,\boldsymbol{R}(X_i - C)\right) < \varepsilon. \tag{1}$$

Here $\angle(u, v)$ denotes the angle between vectors u and v. Thus, we are seeking the largest consistent subset of all hypothetical correspondences such that the angular error is less than ε for each correspondence.

An alternative would be to the try to minimize ε such that $|I|$ is larger than some predefined threshold, K, an approach used in [17] for the triangulation problem. Thus we seek the smallest ε such that there is at least K correspondences that satisfy (1). In cases with no outliers, this becomes the standard L_∞ norm formulation.

Problem 2. *For a prescribed $K \in \mathbb{N}$, find the rotation $\boldsymbol{R}$ and position C that solves*

$$\min \varepsilon \quad s.t. \quad |I| \geq K.$$

where I is the set of indices i such that

$$\angle\left(x_i\,,\,\boldsymbol{R}(X_i - C)\right) < \varepsilon.$$

In either formulation, one has to specify a modeling parameter (ε or K). It may seem more convenient to prescribe ε since one usually has some idea of the noise level in the measurements. A side effect of this choice is that there may be a whole set of solutions in the space of Euclidean motions that satisfies (1). This can be regarded as a good feature as one gets uncertainty estimates of the camera positions for free, but in other circumstances, it may be preferable to have a unique solution. From an algorithmic point of view we have found that it is more practical to specify the maximum number of outliers and optimize over ε. We have experimented on both formulations.

3 Pumpkin Constraints

In this section we derive approximate constraints for L_∞ optimality. By considering the angle between two image vectors x and y we get constraints on the camera position that do not involve the orientation.

Assume for a moment that we have no noise and no false correspondences. Then, given two world points, X and Y and the corresponding image vectors x and y, we have

$$\angle\big(X - C, Y - C\big) = \angle\big(x, y\big). \tag{2}$$

This yields a constraint on the camera position C, that we intend to study more closely. On the left hand side of this equation, X and Y are given by our model of the scene. On the right hand side, we have the angle between two image vectors, which is simply a constant that can be calculated from the measured image coordinates. We denote it by α. We seek the points C in space that form exactly the angle α with X and Y.

So for which points does this hold? First consider a plane through X and Y. It is a well known result from classical geometry that if X and Y are two fixed points on a circle, then the angle XCY is equal for all points C on the arc from X to Y. This tells us that the points C such that XCY is equal to α form two circular arcs in the plane as shown in Figure 1. Also note that if for some other camera position $\bar{C}$ the angle $X\bar{C}Y$ is larger than α, then $\bar{C}$ lies in the set enclosed by the two arcs.

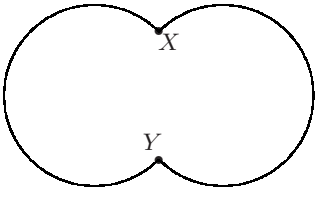


Fig. 1. The points C for which $XCY = \alpha$ form two circular arcs in the plane

In space the points C for which $XCY = \alpha$ form a surface which is obtained by rotating the circular arcs around the line through X and Y (Figure 2). Like in the planar case any $\bar{C}$ such that $X\bar{C}Y > \alpha$ lies in the set enclosed by this surface. We define

$$M_\alpha(X, Y) = \{C \in \mathbb{R}^3 : XCY > \alpha\}.$$

If $\alpha < \pi/2$, this set will be non-convex and shaped like a pumpkin.

Now assume we have found an optimum $(\boldsymbol{R}, C)$ in the sense given by Problem 1 or Problem 2. Let X and Y be two points that satisfy (1). Then by the spherical version of the triangle inequality,

$$\begin{aligned}\angle\big(X - C, Y - C\big) &\leq \alpha + 2\varepsilon \\ \angle\big(X - C, Y - C\big) &\geq \alpha - 2\varepsilon.\end{aligned}$$

Note that these constraints are weaker than the L_∞ norm, so they could be used to produce tights bounds on the L_∞ optimum. We propose a branch and bound algorithm, evaluating these constraints for smaller and smaller boxes.

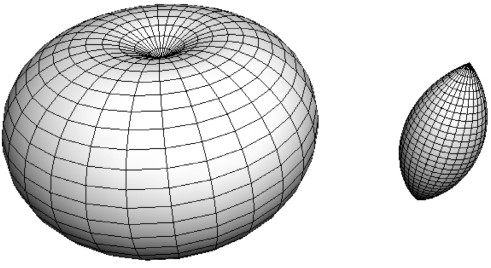


Fig. 2. M_α for angles less than (*left*) and larger than (*right*) $\pi/2$ respectively

4 An Algorithm

The constraints from the previous section have the important characteristic that they do not involve the orientation of the camera. Thus we can seek the optimal camera centre by a branch and bound search over a subset of $\mathbb{R}^3$. For each camera centre we evaluate the pumpkin constraints and get a bound on the L_∞ norm of this specific position. As we will show later, evaluating constraints of this type can be done quickly so there is chance of a reasonably fast algorithm. More importantly, the fact that each constraint depends only on two correspondence pairs makes it is possible to handle outliers. Suppose, for example, that a certain camera centre violates the constraints from k disjoint pairs (X_i, X_j). Then a solution with this camera centre will have at least k outliers.

We will now present an algorithm to estimate the optimal camera position with respect to any of the error measures presented in Section 2. Starting with a rough lower bound on the optimum, we use a branch and bound approach to create tighter and tighter bounds on the optimal solution. The aim is to restrict the solution to a small enough volume in space. The box below presents the overall structure of the algorithm and we will now go through the main steps in greater detail.

Algorithm 1

Initialize to get a bounded set in $\mathbb{R}^3$.
Iterate until desired precision is reached:
1. *Pick a box from the queue.*
2. *Evaluate all pumpkin constraints for this box.*
3. *Try to detect and remove outliers.*
4. *Try to discard the box.*
5. *If the box cannot be discarded:*
   - *Divide the box and update the queue.*
   - *Try to update the lower bound on the optimum.*
6. *Remove the box from the queue.*

Initialization. To start our branch and bound loop, we need a bounded set of possible camera centres. In many cases we can get this from our knowledge of

the scene. For example, we might know the size of the building we are working with. Otherwise we propose an initialization scheme by considering the pumpkin constraints directly on sets of the kind $|C_i| > b$ where C_i is one of the coordinates of the camera centre. Since for most pumpkin constraints $\alpha > 0$, they will be bounded sets in space and thus there will be a maximal b such that they intersect $|C_i| > b$.

Evaluating the constraints. Consider two world points X and Y and their corresponding image points x and y. If the angle $\alpha = \angle(x, y)$ is smaller than $\pi/2$, the constraint from Section 3 will be non-convex. So how do we determine whether any part of a given box (in our branch and bound algorithm) intersects this pumpkin?

Remember the way we derived the pumpkin constraints, a circular arc was rotated around the line through X and Y. Now consider the curve formed by the circle centre when rotating. This is itself a circle centred around $(X + Y)/2$ and the pumpkin is simply all points in space with less than a certain distance to this circle. To check if a given box intersects such a pumpkin we inscribe the box in a sphere and calculate the shortest distance from the sphere to the central circle of the pumpkin, see Figure 3. We know that the vector from the centre of the sphere to the closest point on the circle is perpendicular to the circle at this point. Thus we can find the shortest distance by studying a plane through the centre of the sphere and the centre of the circle (being $(X+Y)/2$) that intersects the circle under a right angle. A similar discussion tells us how to handle the case $\alpha > \pi/2$.

Detecting Outliers and Discarding Boxes. Though the method we present can compete with other optimal methods when it comes to speed, a greater advantage is the ability to deal with outliers. The basic idea is very simple. If there are no outliers, we only need to find one violated constraint to discard a box in the branch and bound. In the case of outliers we need to find a larger number of violated constraints.

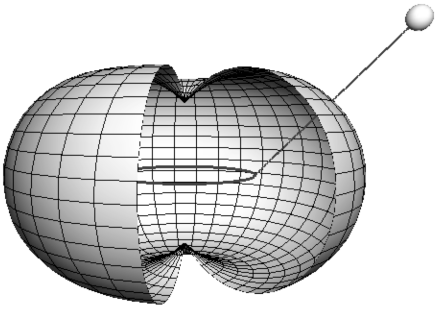


Fig. 3. The central circle inside of M_α centred around $(X+Y)/2$ and the line segment between the central circle and the sphere (circumscribing a box)

Actually, it is often even possible to pinpoint which correspondences are the outliers. Suppose a certain point is involved in, say, 10 different violated constraints. If this point were to be an inlier, then the other 10 points would have to be outliers. In this manner it is possible for each box in the branch and bound queue to keep track of which correspondences one needs to consider.

Updating the Lower Bound. To detect outliers and discard boxes in the branch and bound algorithm, we need some lower bound on the number of inliers of the optimal solution. At the initial stage this could be an educated guess or the result of an approximate method. For example one could use a RANSAC solution as a lower bound on the optimal solution. However, as the algorithm progresses we need to update this bound. In the next section we will show how to do this formally with respect to the error measures of Section 2. Here, we give an approximate method that works well in practice.

Consider a box that could not be eliminated in step 4 of Algorithm 1. A hypothesis is that the optimal camera position lies close to the centre of this box. Following this idea, we fix the camera position to the centre of this box and estimate the camera rotation. Since we have already eliminated most of the outliers we can estimate the rotation (for the remaining points) with Horn's closed form solution using unit quaternions [20]. Now when we have fixed both camera position (to the box centre) and rotation (to the Horn solution) we estimate the error as in Section 2. This gives us a new lower bound on the optimal solution.

5 Optimal Rotation

In the previous section, we presented a practical approach to find the camera centre that is optimal with respect to the L_∞ norm of the reprojection errors. It works to find smaller and smaller sets which are guaranteed to contain the optimal solution. We say that the algorithm has converged when a set is found which is small enough for the user's needs. However, Algorithm 1 is not guaranteed to converge to any prescribed precision. In this section, we discuss how to securely find the optimum by searching the space of rotations, whence giving sufficient conditions for obtaining globally optimal solutions.

It should be noted that in our experiments, the basic Algorithm 1 produces the desired precision and solves the optimization problem. Thus the modifications of this section are only required to guarantee convergence. Though we have implemented and tested Algorithm 2, the performance in Section 6 are from using Algorithm 1 with Horn's method to estimate rotations.

Searching Rotation Space. We will use the same approach to refine two different steps in Algorithm 1, discarding boxes in step 4 and updating the lower bound on the optimum in step 5. We will mainly describe the second problem here, but the same approach can be used for the first problem. Just as in the previous section, we start with a box that could not be discarded. To get a lower

bound on the optimum, we fix the camera position to the centre of this box. Assuming that we use the formulation from Problem 1, we want to solve

Problem 3. *Given a camera position $\tilde{C}$ and $\varepsilon > 0$, find the rotation $\boldsymbol{R}$ that maximizes $|I|$, where I is the set of inliers as defined in Problem 1.*

Algorithm 2

Start with a set of spherical triangles covering the sphere.
Iterate until desired precision is reached:
1. *Pick a triangle from the queue.*
2. *Try to discard the triangle.*
3. *If the triangle cannot be discarded:*
   - *Divide the triangle and update the queue.*
   - *Try to update the lower bound on the optimum.*
4. *Remove the triangle from the queue.*

Our measured image is represented with unit vectors x_i. Since we have fixed the camera centre to $\tilde{C}$ and have a 3D model of the scene we can calculate a modeled image, i.e. what the image should look like, given the 3D model and the fixed camera centre. We represent the modeled image with unit vectors

$$\tilde{x}_i = (X_i - \tilde{C})/||X_i - \tilde{C}||.$$

Problem 3 consists in finding the rotation that maps as many of the $\tilde{x}_i$'s as possible to the corresponding x_i's, within the prescribed tolerance ε. As a parameterization for rotations we use a unit vector ζ and an angle γ. The unit vector specifies the point in the modeled image that will be mapped to the z axis of the measured image and γ is the rotation angle around this axis.

To find the optimal rotation, we propose a branch and bound search over the possible ζ's. The unit sphere is divided into spherical triangles. A simple test is then used to determine if a given triangle can contain the optimal solution, if so it is divided into four new triangles. Figure 5 shows the evaluated triangles for one search.

Discarding Triangles. It remains to discuss how to discard triangles in the branch and bound search. Given a (spherical) triangle of possible ζ's we want to determine if the optimal solution can lie within this triangle. Let ζ_c be the centre of the triangle and ρ the largest distance from the centre to a point in the triangle. It is easy to show that the reprojection errors are Lipschitz continuous as functions of ζ. This implies that if the optimal rotation is in the considered triangle mapping k points with error less than ε, then there is a rotation with $\zeta = \zeta_c$ that maps k points with error less than $\varepsilon + \rho$. Thus we assume that $\zeta = \zeta_c$ and look for a γ such that as many points as possible are mapped with error less than $\varepsilon + \rho$.

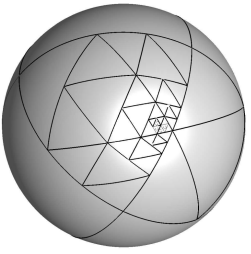


Fig. 4. Plot of the considered triangles in a particular search over the sphere. The data used are from the Notre Dame data (see Section 6).

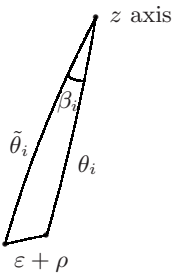


Fig. 5. The figure shows the modeled image superposed with the measured. The lower triangle corners are the modeled image point and the measured respectively. Their distance must not be larger than $\varepsilon + \rho$.

First transform all image points $\tilde{x}_i$ in the modeled image to the measured with a rotation R that maps ζ to the z axis. Now that we have all points in the same coordinate system we switch to spherical coordinates

$$x_i = \left(\sin\theta_i \cos\phi_i \ , \sin\theta_i \sin\phi_i \ , \cos\theta_i \right)$$

and similarly for the modeled image points. Note that θ_i is the angle between the image point and the z axis. The γ-rotation will just add to the ϕ-angles of the modeled image points. Thus, if there were no noise or errors we would have $\theta_i = \tilde{\theta}_i$ and $\phi_i = \gamma + \tilde{\phi}_i$ for all i's. In practice this will not be the case. Instead we seek γ such that our error measure is minimized. This is equivalent to

$$|\tilde{\phi}_i + \gamma - \phi_i| < \beta_i \tag{3}$$

for as many i's as possible. The angle β_i depends on the tolerance ε and the triangle size ρ. Figure 5 illustrates how to calculate β_i using the spherical law of cosines.

We have thus to decide a γ that satisfies (3) for as many i's as possible. Since the constraints are intervals this is rather straightforward and we will not go into the details here. The maximal number of satisfied constraints gives us an upper bound on the number of inliers of the best solution in the current triangle. If it is lower than the number of inliers of the best solution so far, we can discard this triangle.

Updating the Lower Bound. If a triangle cannot be discarded as above, we try to update the lower bound on the optimum. As a candidate we use $\zeta = \zeta_c$, being the centre of the triangle that we could not discard. Then we repeat the procedure above, but without the extra tolerance ρ.

6 Experiments

To make it easier to compare the performance of our method to existing ones we have tested it on data without as well as with outliers. The timings presented are for a simple C++ implementation on a 1.2 GHz iBook G4.

For some data sets, like the Notre Dame scene described below, the size of the different scenes varies dramatically. To get a reasonable measure of performance we have normalized the size of the different scenes so that both 3D points and the camera centre fit into a into a $10 \times 10 \times 10$ voxel cube. The same normalization was done on the dinosaur data set.

Dinosaur Data. As a first practical experiment of our algorithm, we ran it on the well-known and publicly available dinosaur images for which a 3D reconstruction of the object is available. In total the data consists of 328 object points and 36 images. Between 22 and 154 points are visible in each view and there are no outliers.

For all 36 views of the dinosaur data we estimated the camera pose. The maximal reprojection errors of the presented solutions are between 0.0003 and 0.001 radians. The median computation time was 0.7 s and the maximal computation time 4.8 s. In Figure 6, the resulting camera trajectory is plotted. As a comparison, we have included the result using standard bundle adjustment of the L_2 error cost function. Note that the two camera trajectories differ little which could be expected as the geometry is not very challenging.

Notre Dame Data. We also tested our algorithm on the Notre Dame data set (which was used in [21]) consisting of 595 images with computed 3D scene structure. From each of the 595 images we picked sets of between 6 and 30 points on which we tested our algorithm. We ran a fixed number of iterations and measured the remaining volume in the space of camera positions. In 96% of the cases the remaining volume of camera positions was less than 0.01. The median computation time was 0.6 s but since some images took quite a long

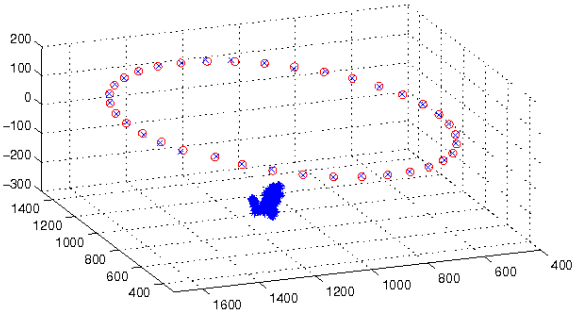


Fig. 6. Estimated camera motion for the dinosaur data using our approach (blue x's) and bundle adjustment (red o's)

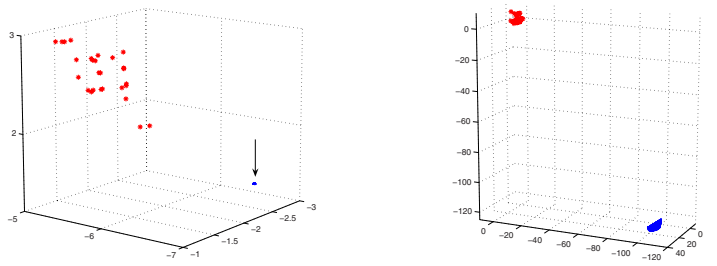


Fig. 7. Two examples from the Notre Dame data. The left plot shows 3D points (red *'s) and feasible camera positions (blue) in a standard case. The right plot shows one of the difficult cases with slower convergence.

time the average computation time was as much as 4.5 s. In the cases where we were not so successful in terms of execution times the point configurations were almost degenerate. The camera was far away from the scene compared to the size of the scene. Figure 7 shows the image points and the calculated camera positions in two cases, one typical scene and one almost degenerate.

Toy Truck Data. We also did some testing on real data with outliers. Figure 8 shows one of 18 images of a scene with a toy truck. We used two of the views to calculate a 3D model of the scene and then matched the other images to this model to get hypothetic correspondences for our experiment. Matching hypotheses were obtained with SIFT descriptors. The number of matched points in each case varied between 23 and 60 and the amount of outliers was around 20%. The average execution time was 8.47 s and the average remaining volume was 0.0005 compared to a scene of size approximately $1 \times 1 \times 1$.

Synthetic Data. To get a larger amount of data with outliers we also evaluated our algorithm on synthetic data. A set of 50 random 3D points were scattered in

Fig. 8. Example of a real case with outliers

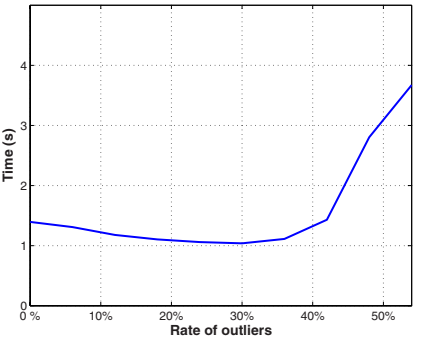


Fig. 9. Mean run times for the synthetic experiments with outliers

a cube with side length 10 units. The camera was placed at distance 10 from the cube. Image vectors were computed and random angular noise with truncated normal distribution (original standard deviation 0.0015, truncated at 0.002) was added to the image vectors. Then a number of the 3D points were exchanged for outlier data with the same distribution. In this case the formulation from Problem 2 was used. The algorithm was run until the remaining volume was less than 0.001. Figure 9 shows the mean run times over 50 experiments for different rate of outliers.

The results indicate that our approach is applicable even with a considerable amount of outliers and that the execution times are competetive. It might seem strange that the run times in Figure 9 initially decrease with increasing rate of outliers. However, since all examples have the same total number of correspondences, the examples with many outliers get easier as the outliers are eliminated. The experiment shows that in this setting and with reasonable rates of outliers, removing the outliers is not the most time-consuming task.

7 Conclusions

The proposed algorithm is a first step towards practical L_∞ pose estimation with outliers. Experiments demonstrate the validity of the approach by computing globally optimal estimates while discarding outliers at the same time. We also believe that this work can be generalized to other problems in computer vision, especially multiview geometry problems. For instance, exactly the same idea can be applied to the triangulation problem and this gives us immediately an optimal algorithm for outlier removal.

The speed of the present algorithm is already attractive for many practical vision problems, but there is still work to be done in this direction for handling a larger percentage of outliers. Sorting the order of the pumpkin constraints that are processed in the branch and bound process based on violation performance will improve speed since non-interesting boxes are cut off early. Another research direction we intend to pursue is the possibility of a GPU implementation as the algorithm is easily parallelizable.

Acknowledgments

This work has been funded by the European Commission's Sixth Framework Programme (SMErobot grant no. 011838), by the European Research Council (GlobalVision grant no. 209480) and by the Swedish Foundation for Strategic Research through the programme Future Research Leaders.

References

1. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
2. Atkinson, K.B.: Close Range Photogrammetry and Machine Vision. Whittles Publishing (1996)

3. Grunert, J.A.: Das pothenot'sche problem in erweiterter gestalt; nebst bemerkungen über seine anwendung in der geodäsie. Grunert Archiv der Mathematik und Physik 1(1841), 238–248
4. Olson, C.: A general method for geometric feature matching and model extraction. Int. Journal Computer Vision 45, 39–54 (2001)
5. Cass, T.: Polynomial-time geometric matching for object recognition. Int. Journal Computer Vision 21, 37–61 (1999)
6. Jacobs, D.: Matching 3-d models to 2-d images. Int. Journal Computer Vision 21, 123–153 (1999)
7. Huttenlocher, D., Ullman, S.: Object recognition using alignment. In: Int. Conf. Computer Vision, London, UK, pp. 102–111 (1987)
8. Jurie, F.: Solution of the simultaneous pose and correspondence problem using gaussian error model. Computer Vision and Image Understanding 73, 357–373 (1999)
9. Breuel, T.: Implementation techniques for geometric branch-and-bound matching methods. Computer Vision and Image Understanding 90, 258–294 (2003)
10. David, P., DeMenthon, D., Duraiswami, R., Samet, H.: SoftPOSIT: Simultaneous pose and correspondence determination. Int. Journal Computer Vision 59, 259–284 (2004)
11. Haralick, R.M., Lee, C.N., Ottenberg, K., Nolle, M.: Review and analysis of solutions of the 3-point perspective pose estimation problem. Int. Journal Computer Vision 13, 331–356 (1994)
12. Quan, L., Lan, Z.: Linear $n \leq 4$-point camera pose determination. IEEE Trans. Pattern Analysis and Machine Intelligence 21, 774–780 (1999)
13. Hartley, R., Kahl, F.: Optimal algorithms in multiview geometry. In: Asian Conf. Computer Vision, Tokyo, Japan (2007)
14. Olsson, C., Kahl, F., Oskarsson, M.: Optimal estimation of perspective camera pose. In: Int. Conf. Pattern Recognition, Hong Kong, China, vol. II, pp. 5–8 (2006)
15. Hartley, R., Kahl, F.: Global optimization through searching rotation space and optimal estimation of the essential matrix. In: Int. Conf. Computer Vision, Rio de Janeiro, Brazil (2007)
16. Sim, K., Hartley, R.: Removing outliers using the L_∞-norm. In: Conf. Computer Vision and Pattern Recognition, New York City, USA, pp. 485–492 (2006)
17. Li, H.: A practical algorithm for L_∞ triangulation with outliers. In: Conf. Computer Vision and Pattern Recognition, Minneapolis, USA (2007)
18. Olsson, C., Enqvist, O., Kahl, F.: A polynomial-time bound for matching and registration with outliers. In: CVPR 2008 (2008)
19. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. Commun. Assoc. Comp. Mach. 24, 381–395 (1981)
20. Horn, B.K.P.: Closed-form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America A 4 (1987)
21. Snavely, N., Seitz, S., Szeliski, R.: Photo tourism: Exploring photo collections in 3d. ACM SIGGRAPH 25, 835–846 (2006)

Learning to Recognize Activities from the Wrong View Point

Ali Farhadi[1] and Mostafa Kamali Tabrizi[2]

[1] Computer Science Department, University of Illinois at Urbana Champaign
afarhad2@uiuc.edu
[2] Institute for Studies in Theoretical Physics and Mathematics
kamali@ipm.ir

Abstract. Appearance features are good at discriminating activities in a fixed view, but behave poorly when aspect is changed. We describe a method to build features that are highly stable under change of aspect. It is not necessary to have multiple views to extract our features. Our features make it possible to learn a discriminative model of activity in one view, and spot that activity in another view, for which one might poses no labeled examples at all. Our construction uses labeled examples to build activity models, and unlabeled, but corresponding, examples to build an implicit model of how appearance changes with aspect. We demonstrate our method with challenging sequences of real human motion, where discriminative methods built on appearance alone fail badly.

1 Introduction

Human activity recognition is a core unsolved computer vision problem. There are several reasons the problem is difficult. First, the collection of possible activities appears to be very large, and no straightforward vocabulary is known. Second, activities appear to compose both across time and across the body, generating tremendous complexity. Third, the configuration of the body is hard to transduce, and there is little evidence about what needs to be measured to obtain a good description of activity.

There is quite good evidence that many activities result in strong spatio-temporal patterns of appearance, and most feature constructions are based on this observation. However, such constructions are bound to a view — if the camera rotates with respect to the body, a disastrous fall in discriminative performance is possible (see Section 6). One strategy is to learn a new model for each view; but it isn't practical to obtain several examples of each activity in each view. An alternative, which we expound in this paper, is to *transfer* models between views. We do so by building models in terms of features which are stable with change of view, yet discriminative.

2 Background

There is a long tradition of research on interpreting activities in the vision community (see, for example, the extensive surveys in [18,20]). Space allows touching only on most relevant points.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 154–166, 2008.

Appearance features are widely used. Generally, such features encode (a) what the body looks like and (b) some context of motion. At low spatial resolution when limbs cannot be resolved, flow fields are discriminative for a range of motions [3]. At higher resolutions, appearance features include: braids [27]; characteristic spatio-temporal volumes [7]; motion energy images [9]; motion history images [9]; spatio-temporal interest points [23]; nonlinear dimensionality reduced stacks of silhouettes [38]; an extended radon transform [40]; and silhouette histogram of oriented rectangle features [21].

Primitives: Motions are typically seen as having a compositional character, with hidden Markov models used to recognize, among others: tennis strokes [45]; pushes [43]; and handwriting gestures [46]. Feng and Perona [17] call actions "movelets" and build a vocabulary by vector quantizing a representation of image shape. These codewords are then strung together by an HMM, representing activities; there is one HMM per activity, and discrimination is by maximum likelihood. Alternatively, Bregler fits a switching linear dynamical system ([10]; see also [37]); Ikizler and Forsyth build primitives explicitly by clustering motion capture [21].

Aspect: Appearance features are subject to aspect problems, which are not generally studied. One alternative is to model in 3D. There is a strong evidence that 3D configuration can be inferred from 2D images (e.g., [19,6,33]; see also discussion in [18]). At higher spatial resolutions one can recover body configuration and reason about it [31,21], and, though inferred body configuration can be quite noisy, there is some evidence this leads to aspect invariant methods [21], or learn in 3D and recognize in 2D [41]. Alternatively, one might use an implicit representation of 3D shapes as a set of silhouettes from multiple views [2,26], spatio-temporal volumes derived from such a silhouettes [42,47], or spatio-temporal curvature of 2D trajectory [32]. Things are more difficult when only one view is available, on which to compute feature values. View invariant motion representations offer one solution, but one must use model-based invariants [29].

Transfer Learning: Transferring knowledge across related tasks is a known phenomenon in human learning [11]. In machine learning, simply pooling the training data does not necessarily work because decision boundaries in the feature space may not be similar for similar tasks. For example, in Figure 2 corresponding frames in camera 1 and camera 4 are totally different.

An extensive literature review is available at [22]. To summarize, two main categories of approach have been used in the literature. One is to use models that are robust under change of domain. One might: use similar tasks to reasonably estimate the prior [48,30,**?**]; use hierarchical bayesian models with hyper priors constrained to be similar for similar tasks [5,25,44]; consider a regularized multi task learning framework, when there is a trade-off between large margins and similar decision boundaries [15]; boost the prediction power of the most helpful out of domain data [12]; or transfer rules in a reinforcement learning framework [34,44].

The second category of approach is to design features which are well behaved under change of domain. The main idea is to have a common feature mapping in which similar objects in different domains look similar, and dissimilar objects look different, even in the same domain. To do this, one might introduce some auxiliary artificial tasks to

uncover the shared structure among similar tasks [4], or learn a distance function which behaves well under transfer [35].

Most similar to our work is the approach of Farhadi, Forsyth and White [16], who construct the space of comparative features. These features compare new words (in sign language) with base words for a view, and in turn, use the result of the comparison as a feature.They demonstrate that, once a semantically similar feature space has been constructed, models of sign language words learned from an animated dictionary alone can be used to recognize sign language words produced by a human signer in different aspects.They claim that since comparisons are transferable, the comparative features will be well behaved under change of domain. We believe that the random searches on the split space,used in [16], result in a noisy feature space. This, we think, is the main reason they needed a topic model on the top of their comparative features. We differ from their work in that: we have a more focused search mechanism for finding stable features and consequently we don't need to have a topic model; and our process appears to apply quite generally, rather than in the specific domain of ASL.

Many vision problems are naturally seen as transfer learning problems (for example, the standard problem of determining whether two face images match without ever having seen images of that individual; as another example, one might use cartoons to learn the location of object features, and very few real images to learn their appearance [14]).

3 Activity Model

We first describe frames of activities and their context. We then look at the long timescale. We describe frames by vector quantization of local appearance features to form the codewords. We match based on the frequencies with which codewords appear in a sequence. In the object recognition literature, local texture codewords are strongly discriminative [13], without spatial information. Similarly, we do not represent precise temporal orderings. Quantitative evaluations show strong discriminative performance for these activity descriptors. See Table 2 and 3 for detailed accuracies.

Descriptive Features: We employ the feature extraction technique of [36]. Their descriptor is a histogram of the silhouette and of the optic flow. Given the bounding box of the actor and the silhouette, the optic flow is computed using Lucas-Kanade algorithm [24]. They consider using a normalized size feature extraction window. Features consist of three channels, smoothed horizontal flow, smoothed vertical flow, and the silhouette. The normalized feature extraction window is divided into 2×2 sub-windows. Each sub-window is then divided into 18 pie slices covering 20 degrees each. The center of the pie is in the center of the sub-window, and the slices do not overlap. The values of each channel are integrated over the domain of every slice. The result is a 72-dimensional histogram. By concatenating the histograms of all 3 channels we get a 216-dimensional frame descriptor.

To encode local temporal structure of the activities we consider stacking features from previous and next frames. We pick the first 50 principal components of the descriptors of a window of size 5, centered at the frame we want to describe. For further frames in both directions, previous and next frames, we pick the first 5 principal components of the windows of size 5, centered at the $(i+5)^{th}$ and $(i-5)^{th}$ frames. This gives us a 60 dimensional descriptor for each frame, which we call descriptive features.

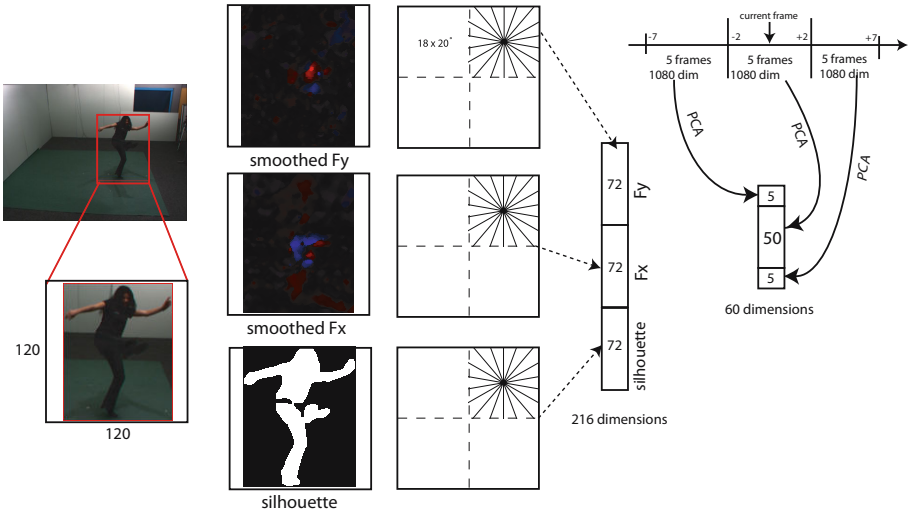


Fig. 1. Feature Extraction: Each frame is described by silhouette information, horizontal and vertical optical flow on a normalized size feature extraction window. We histogram all three channels of information by integration over angular slices of subwindows of the bounding box. We then stack information about previous and next frames to model local dynamics of activities.

Vector quantization: There is evidence that motions are composites of short timescale primitives [17]. Temporal order over short to medium timescales does not seem to be an important discriminative cue, so we can use a bag of features model. This suggests vector quantizing frame descriptors to form codewords. We use K-means clustering.

Matching: Each activity is described by a histogram of codewords, which we call codeword-based description. We use a 1 Nearest Neighbor (1NN) classifier to recognize an activity. We use hamming distance for matching the histograms.

Figure 1 depicts the feature extraction procedure. These features are discriminative, 98.92% average accuracy on Weizman10 dataset [8] is reported in [36] using a 1NN classifier on these features, comparing to 97.78% average accuracy of [39] and 82.6% average accuracy of [28] on the same dataset.

4 Transferable Activity Model

We consider two types of activities: *shared* activities, that are observed in both *source* and *target* views, and *orphan* activities that are observed only in the source view. Shared activities in both the source and the target view are not labeled, while orphan activity labels are available in the source view. We would like to transfer orphan activity models from the source to the target view. This means that we would like to learn a model for an orphan activity in the source view and test it in the target view.

Activities look different from different view points. Each column in Figure 2 shows how much a particular frame in an activity looks different across view points. This

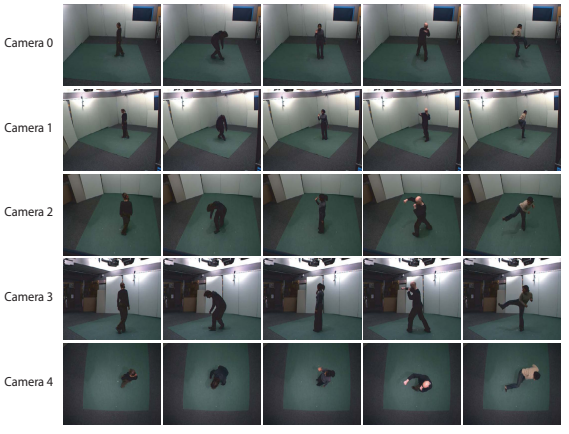


Fig. 2. The IXMAS dataset, Each column shows changes across the views. Because of these differences learning a model of descriptive features in one view and testing it in another view fails badly. Camera 4 has a weird configuration, introducing ambiguities in action classification for this view.

suggests that, under transfer circumstances, appearance based description will work poorly. It does; a 1NN classifier gets 14% average accuracy (leave one action out) when trained on one camera and tested in another one (for detailed accuracies see Table 2).

To be able to transfer activity models from the source to the target view, we need discriminative features that tend to be similar in different views. Codeword-based representation of activities inside each view seems to be discriminative. We could cluster both the source and the target views. But this is not enough because we do not know which cluster in the target view corresponds to which one in the source view, so we could not transfer a model. We need a method to describe each view such that correspondence is straightforward. For this reason, we build a cell structure on top of the clusters in the source view.

Shared activities are valuable because they demonstrate what the body will look like in the target view, for a given appearance in the source view. Our strategy is to build a discriminative problem around the clusters in the source view. We construct a structure of cells within which clusters lie. These cells are produced by an arrangement of hyperplanes, which we call *splits*. We use the shared activities to train this discriminative task. For shared activities we *choose* which side of each hyperplane in the source view a frame lies. We can now force the corresponding frame in the target view to be on the same side of the corresponding hyperplane in the target view. Figure 3 shows the procedure for making these new features, which we call *split-based* descriptors. If we have enough shared activities, we can establish correspondences between views.

The core idea here is that splits are transferable because we learn them to be. This means that the split-based features of a frame in the source view are similar to the split-based features of the corresponding frame in the target view. As a result, an orphan activity model can be applied to both views, if it is built using split-based features.

4.1 Building Splits in the Source View

We start by clustering the source view to form the codewords. We can now describe activities in the source view by a histogram of the codewords. Our split-based features take the form

$$f_i(x_j^v) = sign(\alpha_i^v \cdot \Phi(x_j^v)). \quad (1)$$

Where $f_i(x_j^v)$ is the i^{th} component of the feature vector of the j^{th} frame in view v, α_i^v is the i^{th} split coefficient in view v and $\Phi(x_j^v)$ is the appearance feature for the j^{th} frame in view v . We would like these features to be discriminative, in the sense that different clusters in one view should have distinct features. These features should also be transferable, meaning that

$$f_i(x_j^s) = f_i(x_{\hat{j}}^t). \quad (2)$$

Where $\hat{j}$ in the target view, is the corresponding frame to frame j in the source view. When we see activities simultaneously from two different views, we get temporal correspondences between frames in different views for free. We can now choose split-based features that are discriminative for the source view. We expect these features, and so orphan models, to transfer because we can produce training data to train on the target view. Training data is the appearance feature in the target view and training labels are, in fact, the split-based descriptors in the source view.

How to choose splits? A random choice of splits was used in [16]. There is a better way. Generally, clusters form blobby structures in the feature space, so cluster members are usually close together. We want splits to form a cell structure that respects the clusters. This implies that splits that do not break up clusters are better. This suggests choosing splits that have large margins with respect to cluster centers. We obtain them by Maximum Margin Clustering (MMC).

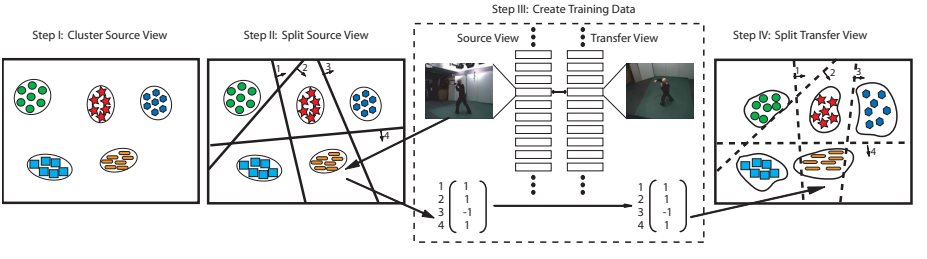


Fig. 3. We first cluster the source view using the original descriptive features. We then use Maximum Margin Clustering to choose informative splits in the source view. Split-based features are constructed by checking which side of each split a particular frame lies. We can directly transfer the split values to the corresponding frames in the target view. Using unlabeled shared activities, which have established implicit correspondences, splits are transferred from the source to the target view. We now can learn splits in the target view, using the descriptive features as training data and transferred split values as labels. These split-based features are transferable, and one could simply use an already learned model in the source view to recognize orphan activities in the target view.

Feature selection is a problem here. Similar to [16], we use several different random projections of the data. We apply MMC to each random projection to get several splits. Different projections usually yield different splits. We discard redundant splits and score the remainder by how well they can be predicted from the data.

Max Margin Clustering. For each random projection we look for the maximum margin hyperplane separating cluster centers in the source view. This lowers the computational burden of finding splits.

One can find these maximum margin hyperplanes by solving a Semi-Definite Programming problem, which is computationally expensive. However, [49] introduce an alternating optimization approach on the original non-convex problem of:

$$\begin{aligned} \min_y \min_{\omega,b,\xi_i} \; & \| \omega \|^2 + 2C\xi^T e \\ s.t. \quad & y_i(\omega^T \varphi(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0 \\ & y_i = \{\pm 1\}, -\ell \leq e^T y \leq \ell \end{aligned} \tag{3}$$

where ω, b are SVM parameters, ξ is the slack variable, φ is the mapping induced by the kernel, $C > 0$ is the regularization parameter, y_i is the label for x_i, $\ell \geq 0$ is a constant controlling the class imbalance and e is the vector of ones. There, one fixes y and optimizes over ω, b, ξ then fixes ω, b, ξ and optimizes over y, and so on. They suggest using SVR with the Laplacian loss, instead of SVM with the hinge loss, in the inner optimization subproblem, to avoid premature convergence.

4.2 Building Splits in the Target View

So far, we have chosen different splits of the data in the source view. Since we want the features to be similar in different views, we directly transfer the splits in the source view to the target view. For each random projection, we learn a classifier in the target view. We use appearance features in the target view as training data and split values, transferred from the source view, as labels for training. Figure 3 depicts the whole procedure for constructing the split-based features.

4.3 Natural Transfer of Activity Models

At this point we know the splits in the source view and we have learned them in the target view, so we can describe each frame using the split-based features in both views. For shared activities, the split-based features are the same in both views. For a given orphan activity in the target view, we first construct the split-based features. To do this, we project the original features, using the same random projections used in learning each split. Then, by checking which side of each hyperplane each frame lies we can form the split-based features. Split-based features, as depicted in Figure 3, are binary features explaining which side of each split a particular frame lies.

With enough training data, we will have similar split-based features for orphan activities in both views. This means that, we can simply use the model learned on the source view and test it in the target view. Note that the orphan activities have not ever been observed in the target view. See Table 1 for the general framework on building transferable models.

Table 1. General framework of learning transferable models. One can choose his/her choice of descriptive features and classifier and plug them in to this general framework to build transferable models. All one needs is a pool of unlabeled, but corresponding, data in both views and labeled data for orphan objects only in the source domain.

General framework for building transferable models :
Learning:
1. Extract suitable descriptive features for your problem.
2. Select your choice of classifier.
3. Choose splits using the MMC on random projections of descriptive features in the source domain.
4. Transfer the splits to the target domain, using an established correspondences between unlabeled shared objects in both domains.
5. Learn the transferred splits in the target domain
Recognition:
1. Extract descriptive features for a query orphan object.
2. Construct the split-based descriptors in the target view, using the already learned splits in the target view.
3. Recognize the query orphan object using the model that you learned in the source domain.

5 Experimental Setup

To examine the performance of our transferable activity model we need to choose a dataset with multiple views of multiple actions. We picked the IXMAS dataset [42] because there are 5 different views of 11 actions performed 3 times by 10 actors, sequences are time aligned, and silhouette information is also provided. There is a drawback in using the IXMAS dataset to test our method. The actors were not instructed to have a fixed angle to each cameras. This introduces some noise to our model. Fortunately, there is no large aspect change in each view, probably because the actors needed to see the instructor. See Figure 2 for example frames from the IXMAS dataset.

We try all 20 combinations of transfer among views (there are 5 views in IXMAS dataset). We cluster the source view to 40 clusters using k-means clustering on the descriptive features. We describe activities by histograms of codewords. 1000 different 30-dimensional random projections of descriptive features are used. For each random projection we find the MMC split on the cluster centers in the source view. Redundant splits are discarded and the best 25 splits are picked. Splits are scored based on how well they can be predicted from the data(train set accuracies). We learn splits in the target view using SVMs with Gaussian kernels on $1/10$ of data, due to the computational limitations. This doesn't affect the quality of the splits dramatically. For a given orphan action, we predict which side of each split all frames lie. We then construct the codeword-based description of that action by matching the split-based description of each frame in the target view to the closest one in the source view. We then use a 1NN classifier to predict the action label for the orphan action, using the source view. Due to the limited number of actions in the dataset, we follow the leave-one-action-out strategy for choosing an orphan action. This means that we exclude all examples of the selected orphan action, performed by all the actors, in the target view. We averaged over all possible combinations of selecting orphan actions to report the average transfer accuracy.

Table 2. Multi Class Transfer Results: The row labeled FO gives the accuracy with which activities are discriminated when both trained and tested on that camera. Columns give the result of testing on the camera heading the column (in bold, the target), when trained on the camera given in the row (normal, the source). There are two cases: first, simply training on appearance features (WT) and second, using our transfer mechanism (Tr). For example, training on camera 4 and testing on camera 0 gives an accuracy of 12% if one uses appearance features only, and 51% if one uses our method. Notice that our transfer mechanism significantly improves performance; there is a cost for training on the wrong view, but it is not particularly high. Camera 4 is generally difficult. Accuracy numbers are computed in leave-one-orphan-activity-out fashion. This means that for a given transfer scenario we average over all possible selections of an orphan activity.

	Camera 0		**Camera 1**		**Camera 2**		**Camera 3**		**Camera 4**	
FO	76		76		68		73		51	
	WT	Tr	WT	Tr	WT	Tr	WT	Tr	WT	Tr
Camera 0	NA	NA	35	**72**	16	**61**	8	**62**	10	**30**
Camera 1	38	**69**	NA	NA	15	**64**	8	**68**	11	**41**
Camera 2	16	**62**	16	**67**	NA	NA	6	**67**	11	**43**
Camera 3	8	**63**	8	**72**	8	**68**	NA	NA	8	**44**
Camera 4	12	**51**	11	**55**	15	**51**	9	**53**	NA	NA

6 Results

We consider two general experiments. Transfer of binary classifiers (Table 3), and transfer of multi class classifiers (Table 2). We demonstrate the possibilities of transferring activity models from one view to another view. One could reasonably use more sophisticated activity models than what we are using in these experiments. What we care about is the price that we are paying for transfer, given a classifier. Note that (a) we are not observing orphan activities in the target domain; (b) we don't know the labels for shared activities in both views (c) we are classifying activities in the target view using a classifier learned on the source view.

Table 2 and 3 show average accuracies in all 20 possible transfer scenarios. For each case we report three different accuracies: 1) Fully Observed (FO) accuracies, in which we use 1NN classifier and all the actions are observed in that view, 2) Without Transfer (WT) accuracies, in which we don't observe orphan activities in the target view and we try to classify them using 1NN classifier on the source view, and 3) Transfer accuracies (Tr).

Transfer is hard, as the "without transfer" (WT) columns in Table 2 show. For example, without transfer learning we get an accuracy of 8% for classifying activities in camera 3 using classifiers learned in camera 1. Transfer learning classifies the activities with the accuracy of 72%. This is a big gain, considering that *our classifier has not ever seen an example of orphan activities in the target view*. If the classifier fully observes the activities in the target view, we get an accuracy of 76%, comparing to 72% accuracy under transfer. Note that transferring from (or to) camera 4 is not as good as the other 4 cameras, because we can't build discriminative enough split-based descriptors. The fully observed accuracies in camera 4 shows that due to the odd geometrical configuration of camera 4 (See Figure 2 for some pictures from camera 4), some of the actions look the same, in that view (also reported in [41]).

Table 3. One vs All Transfer Results: The row labeled FO gives the accuracy with which activities are discriminated when both trained and tested on that camera. Columns give the result of testing on the camera heading the column (in bold, the target), when trained on the camera given in the row (normal, the source). There are two cases: first, simply training on appearance features (WT) and second, using our transfer mechanism (Tr). For example, training on camera 4 and testing on camera 0 gives an accuracy of 32% if one uses appearance features only and 87% if one uses our method. Notice that our transfer mechanism significantly improves performance; there is a cost for training on the wrong view, but it is not particularly high. Camera 4 is generally difficult. Accuracy numbers are computed in leave-one-orphan-activity-out fashion. This means that for a given transfer scenario we average over all possible selections of an orphan activity.

	Camera 0		**Camera 1**		**Camera 2**		**Camera 3**		**Camera 4**	
FO	95		96		93		93		87	
	WT	Tr	WT	Tr	WT	Tr	WT	Tr	WT	Tr
Camera 0	NA	NA	69	**96**	35	**88**	33	**86**	26	**75**
Camera 1	67	**95**	NA	NA	35	**91**	35	**81**	26	**74**
Camera 2	41	**86**	42	**87**	NA	NA	28	**89**	31	**75**
Camera 3	27	**87**	32	**84**	20	**91**	NA	NA	23	**76**
Camera 4	32	**87**	32	**88**	30	**84**	27	**83**	NA	NA

Table 4. View Estimation Results. If we don't know the target view, we can identify the view discriminatively using the descriptive features and a 1NN classifier. This table shows classification results for view classification problem. If the target view is not specified, then we should multiply accuracies in Table 2 and 3 by accuracies in this table. Since these numbers are reasonably close to 1, we shouldn't expect a major change in accuracies.

Camera 0	Camera 1	Camera 2	Camera 3	Camera 4
98	95	96	95	85

Estimating view: Our discussion has assumed that the view is known. This means that we know the source and the target view, for computing accuracies in Table 2 and 3. But it is not crucial to know the target view. We can identify the view discriminatively using the descriptive features and a 1NN classifier. Table 4 shows classification results for view classification problem. If the target view is not specified, then we should multiply accuracies in Table 2 and 3 by accuracies in Table 4. Since the numbers in Table 4 are reasonably close to 1, we shouldn't expect a major change in accuracies.

To our knowledge, transferring activity models between views is novel, and so we can not compare directly with other methods. The most similar experiment is [41]. In this case, learning is in 3D, using 3D exemplars, and recognition is in 2D. Authors report an average accuracy of 57.8%, compared to our average transfer accuracy of 58.1%, on the same dataset. In our experiments we use the same experimental setting as in [41]. Although we appear to have only a tiny lead in classification accuracy, the comparison is rough because, to train, they observe all the activities in all views and we do not. Similarly, it is worth mentioning that view invariant approaches [42,2,26,23,32] need to observe all of the activities and labels in all views. *Here, we neither observe orphan activities in the target view, nor know the labels for shared activities in either views.*

7 Discussion

We have shown that a discriminative appearance model of activity can be trained on one view and tested on a second, different view, *as long as one uses the right features*. Our split-based representation requires knowing the view, but this can be determined by a simple matching procedure. These split-based features can be seen as view invariant features, or as a representation of the *structure shared between different tasks (recognize from a fixed view), using a pool of unsupervised data*.

One could cluster activity frames in each view, then build correspondence between clusters explicitly. However, this might require quite small clusters, because a large volume of feature space in one view might correspond to a small volume in another. Our discriminative strategy for building features has the advantage that it implicitly follows these changes in the underlying feature metric.

Our procedure is implemented using activity data, but we believe that the underlying problem — use data from one aspect to be able to recognize something seen at another aspect — is pervasive in vision. Our solution appears to be quite general: ***if one has corresponding, but unlabeled, data, one can learn and account for the distortion of the feature space caused by changes of view***.

Acknowledgements

The authors would like to thank David Forsyth for his insightful comments and helpful discussions. This work was supported in part by the National Science Foundation under IIS - 0534837 and in part by the Office of Naval Research under N00014-01-1-0890 as part of the MURI program. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation or the Office of Naval Research.

References

1. Niculescu-Mizil, R.C.A.: Inductive transfer for bayesian network structure learning (2007)
2. Aloimonos, Y., Ogale, A.S., Karapurkar, A.P.: View invariant recognition of actions using grammars. In: Proc. Workshop CAPTECH (2004)
3. Mori, G., Efros, A.A., Berg, A.C., Malik, J.: Recognizing action at a distance. In: IEEE International Conference on Computer Vision (ICCV 2003) (2003)
4. Ando, R.K., Zhang, T.: A framework for learning predictive structures from multiple tasks and unlabeled data. J. Mach. Learn. Res. 6, 1817–1853 (2005)
5. Bakker, T.H.B.: Task clustering and gating for bayesian multitask learning. Journal of Machine Learning, 83–99 (2003)
6. Barron, C., Kakadiaris, I.: Estimating anthropometry and pose from a single uncalibrated image. Computer Vision and Image Understanding 81(3), 269–284 (2001)
7. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: ICCV, pp. 1395–1402 (2005)
8. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: ICCV (2005)

9. Bobick, A., Davis, J.: The recognition of human movement using temporal templates. PAMI 23(3), 257–267 (2001)
10. Bregler, C., Malik, J.: Tracking people with twists and exponential maps. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 8–15 (1998)
11. Perkins, G.S.D.N.: Transfer of learning, 2nd edn. International Encyclopedia of Education (1992)
12. Dai, W., Yang, Q., Xue, G.-R., Yu, Y.: Boosting for transfer learning. In: ICML 2007 (2007)
13. Dance, C., Willamowski, J., Fan, L., Bray, C., Csurka, G.: Visual categorization with bags of keypoints. In: ECCV International Workshop on Statistical Learning in Computer Vision (2004)
14. Elidan, G., Heitz, G., Koller, D.: Learning object shape: From drawings to images. In: CVPR 2006, Washington, DC, USA, pp. 2064–2071. IEEE Computer Society, Los Alamitos (2006)
15. Evgeniou, T., Pontil, M.: Regularized multi–task learning. In: KDD 2004 (2004)
16. Farhadi, A., Forsyth, D.A., White, R.: Transfer learning in sign language. In: CVPR (2007)
17. Feng, X., Perona, P.: Human action recognition by sequence of movelet codewords. In: Proceedings of First International Symposium on 3D Data Processing Visualization and Transmission,2002, pp. 717–721 (2002)
18. Forsyth, D., Arikan, O., Ikemoto, L., O'Brien, J., Ramanan, D.: Computational aspects of human motion i: tracking and animation. Foundations and Trends in Computer Graphics and Vision 1(2/3), 1–255 (2006)
19. Howe, N.R., Leventon, M.E., Freeman, W.T.: Bayesian reconstruction of 3d human motion from single-camera video. In: Solla, S., Leen, T., Müller, K.-R. (eds.) Advances in Neural Information Processing Systems 12, pp. 820–826. MIT Press, Cambridge (2000)
20. Hu, W., Tan, T., Wang, L., Maybank, S.: A survey on visual surveillance of object motion and behaviors. IEEE Trans. Systems, Man and Cybernetics - Part C: Applications and Reviews 34(3), 334–352 (2004)
21. Ikizler, N., Forsyth, D.: Searching video for complex activities with finite state models. In: CVPR (2007)
22. Kaski1, S., Peltonen, J.: Learning from Relevant Tasks Only. Springer, Heidelberg (2007)
23. Laptev, I., Lindeberg, T.: Space-time interest points (2003)
24. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stero vision. IJCAI (1981)
25. Rosenstein, L.K.M.T., Marx, Z.: To transfer or not to transfer (2005)
26. Niu, F., Abdel-Mottaleb, M.: View-invariant human activity recognition based on shape and motion features. In: ISMSE 2004 (2004)
27. Niyogi, S., Adelson, E.: Analyzing and recognizing walking figures in xyt. In: Media lab vision and modelling tr-223. MIT, Cambridge (1995)
28. Scovanner, M.S.P., Ali, S.: A 3-dimensional sift descriptor and its application to action recognition. ACM Multimedia (2007)
29. Parameswaran, V., Chellappa, R.: View invariants for human action recognition. In: IEEE Conf. on Computer Vision and Pattern Recognition (2003)
30. Raina, D.K.R., Ng, A.Y.: Transfer learning by constructing informative priors (2005)
31. Ramanan, D., Forsyth, D.: Automatic annotation of everyday movements. In: Advances in Neural Information Processing (2003)
32. Rao, C., Yilmaz, A., Shah, M.: View-invariant representation and recognition of actions. IJCV 50(2), 203–226 (2002)
33. Taylor, C.: Reconstruction of articulated objects from point correspondences in a single uncalibrated image. Computer Vision and Image Understanding 80(3), 349–363 (2000)
34. Taylor, M.E., Stone, P.: Cross-domain transfer for reinforcement learning. In: ICML 2007 (2007)

35. Thrun, S.: Is learning the n-th thing any easier than learning the first? In: NIPS (1996)
36. Tran, D., Sorokin, A.: Human activity recognition with metric learning. In: ECCV (2008)
37. Turaga, P.K., Veeraraghavan, A., Chellappa, R.: From videos to verbs: Mining videos for activities using a cascade of dynamical systems. In: IEEE Conf. on Computer Vision and Pattern Recognition (2007)
38. Wang, L., Suter, D.: Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model. In: IEEE Conf. on Computer Vision and Pattern Recognition (2007)
39. Wang, L., Suter, D.: Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model. CVPR (2007)
40. Wang, Y., Huang, K., Tan, T.: Human activity recognition based on r transform. Visual Surveillance (2007)
41. Weinland, D., Boyer, E., Ronfard, R.: Action recognition from arbitrary views using 3d exemplars. In: ICCV, Rio de Janeiro, Brazil (2007)
42. Weinland, D., Ronfard, R., Boyer, E.: Free viewpoint action recognition using motion history volumes. Computer Vision and Image Understanding (2006)
43. Wilson, A., Bobick, A.: Learning visual behavior for gesture analysis. In: IEEE Symposium on Computer Vision, pp. 229–234 (1995)
44. Wilson, A., Fern, A., Ray, S., Tadepalli, P.: Multi-task reinforcement learning: a hierarchical bayesian approach. In: ICML 2007 (2007)
45. Yamato, J., Ohya, J., Ishii, K.: Recognising human action in time sequential images using hidden markov model. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 379–385 (1992)
46. Yang, J., Xu, Y., Chen, C.S.: Human action learning via hidden markov model. IEEE Transactions on Systems Man and Cybernetics 27, 34–44 (1997)
47. Yilmaz, A., Shah, M.: Actions sketch: A novel action representation (2005)
48. Marx, L.K.Z., Rosenstein, M.T.: Transfer learning with an ensemble of background tasks (2005)
49. Zhang, K., Tsang, I.W., Kwok, J.T.: Maximum margin clustering made practical. In: ICML 2007 (2007)

Joint Parametric and Non-parametric Curve Evolution for Medical Image Segmentation

Mahshid Farzinfar[1], Zhong Xue[2], and Eam Khwang Teoh[1]

[1]School of EEE, Nanyang Technological University, Singapore 639798
[2]Department of Radiology, The Methodist Hospital and The Methodist Hospital Research Institute, Weill Cornell Medical College, Houston, TX 77030, USA
mahs0002@ntu.edu.sg, zxue@tmhs.org, eekteoh@ntu.edu.sg

Abstract. This paper proposes a new joint parametric and nonparametric curve evolution algorithm of the level set functions for medical image segmentation. Traditional level set algorithms employ non-parametric curve evolution for object matching. Although matching image boundaries accurately, they often suffer from local minima and generate incorrect segmentation of object shapes, especially for images with noise, occlusion and low contrast. On the other hand, statistical model-based segmentation methods allow parametric object shape variations subject to some shape prior constraints, and they are more robust in dealing with noise and low contrast. In this paper, we combine the advantages of both of these methods and jointly use parametric and non-parametric curve evolution in object matching. Our new joint curve evolution algorithm is as robust as and at the same time, yields more accurate segmentation results than the parametric methods using shape prior information. Comparative results on segmenting ventricle frontal horn and putamen shapes in MR brain images confirm both robustness and accuracy of the proposed joint curve evolution algorithm.

1 Introduction

Image segmentation or automatically extracting useful information of anatomic structures plays an important role in medical image analysis. In literature, various methods have been proposed for image segmentation using either explicit or implicit curve descriptors. Typical explicit descriptors include generic deformable templates [1,2] and free form active contours [3] wherein points along the curves are used to represent object shapes. In these methods the energy functions or evolution forces are calculated based on the image data along a deformable curve, in order to drive it to the desired location in the image. Implicit curve descriptors are generally parameterized by arc-length [4,5]. One of the typical methods using implicit curve descriptor, known as the level set method [6], has become very important in computer vision due to its advantages such as parameter-free representation of curves and easy handling of topology changes. Active contours based on the level set methods can ideally detect any arbitrary smooth object shapes by minimizing image boundary-based energy functions

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 167–178, 2008.

[4,5,7]. However, they are generally sensitive to image noise and often stuck in local minima when the image contrast is low, since they rely on image boundary features and do not use any shape constraints. Compared to the boundary-based energy functions, region-based energy functions that utilize regional image statistics inside and outside the contours [8,9] often yield relatively robust segmentation results, since regional features are normally more stable than boundary features. However, these region-based energy functions estimate image region features by highly restricted shape models, and they can not guarantee correct segmentation of object shapes, especially for partly occluded images. Therefore, similar to active contours, using both region and boundary-based image features in level set models is often inadequate to segment complex and variable object.

On the other hand, deformable template or model-based shape matching methods have received more attention to process globally high-level information of object shapes. One important property of the model-based approach is that the variability of object shapes is captured from training samples and acts as the shape priors. The shape prior information can be used to both parametric and non-parametric curve evolution. In parametric models, Cootes *et al.* [1] applied the principle component analysis (PCA) to model the distribution of object shapes. Leventon *et al.* [7] extended geometric active contour evolution [4] by using a statistical shape model. In level set curve evolution, Tsai *et al.* [9] proposed to parameterize the level set energy function with respect to shape and pose parameters, and performed curve evolution parametrically via updating the shape and poses parameters constrained by the statistical parametric model obtained using PCA. In non-parametric models, Chen *et al.* [10] extended geodesic active contours by using coupled shape prior knowledge. They constructed non-probabilistic prior model from the training set of curves. Rousson and Paragios *et al.* [11] proposed a variational method in level set framework which is enhanced by shape information, and the prior shape knowledge is represented by a local Gaussian distribution model. Cremers *et al.* [12] assimilated non-linear statistical shape model derived by performing kernel PCA along level set-based Mumford-Shah segmentation. Cremers *et al.* [13] and Kim *et al.* [14] used non-parametric density estimation for shape priors in the level set framework. In summary, non-parametric level set curve evolution can match image boundaries accurately but often suffer from local minima, and parametric level set curve evolution uses statistical shape models, and it is more robust but less accurate in matching object boundaries.

Integrating the advantages of both parametric and non-parametric level set curve evolution methods, in this paper, we formulate the segmentation problem using a maximum a posteriori (MAP) framework such that the overall energy function consists of both the parametric and the non-parametric curve evolution terms.Since the parametric statistical model-based curve evolution is constrained by the shape priors that reflect the shape variability trained from available samples, the algorithm is robust and less affected by noise or low image contrast. On the other hand, the non-parametric evolution energy term can handle infinite degree of curve variations and helps us to achieve better accuracy than the method

that only uses the parametric statistical models such as Tsai *et al.* [9]. In the optimization procedure of the joint curve evolution, both the parametric and non-parametric energy terms are optimized iteratively. There are also recent works by Rousson *et al.* [15] and Bresson *et al.* [16], which obey the same statistical model in level set method. However, one important difference is that these methods directly apply a PCA model to constrain the level set function while matching it with the image features. Thus, the level set function is more constrained by the prior distribution. On the other hand, our algorithm jointly uses two curves to match the object shape in an iterative manner: one curve is represented by the level set function (non-parametric curve) and another is represented by the statistical model (parametric model), according to the Bayesian framework. They jointly constrained via a similarity measure between them in each iterative evolution. Finally, the non-parametric level set curve is regarded as the final matching result since it is more accurate as compared to the parametric curve. In experiments, the proposed algorithm has been applied to segment the frontal horns of ventricles and putamens of MR brain images. Experimental results show that the proposed joint curve evolution is as robust as Tsai *et al.* method [9], and it also yields more accurate matching results by using manually marked curves as the gold standard.

2 Previous Methods

In this section, we briefly introduce the non-parametric and parametric curve evolution algorithms that the joint curve evolution algorithm is built on.

2.1 Non-parametric Curve Evolution

In [8], Chan and Vese *et al.* proposed a non-parametric curve evolution method by solving the Mumford Shah problem, called minimum partition. From the variational analysis point of view, the basic idea of the algorithm is to find the level set function ϕ of a given image I by minimizing the energy function,

$$E_{img}(\phi, I) = \int_W (I(\mathbf{x}) - c_W)^2 d\mathbf{x} + \int_{\Omega \backslash W} (I(\mathbf{x}) - c_{\backslash W})^2 d\mathbf{x} + \nu \int_\Omega \mid \bigtriangledown H(\phi)\mid, \quad (1)$$

where ν is a constant, $W \subset \Omega$ is an open subset of image domain Ω. The level set function ϕ divides the image domain Ω into two homogeneous regions W and $\Omega \backslash W$. c_W represents the average intensity of image within region W, and $c_{\backslash W}$ is the average intensity within region $\Omega \backslash W$. It can be seen that this non-parametric curve evolution algorithm uses region-based image features and infinite degree of curve variation for image segmentation.

2.2 Parametric Model-Based Curve Evolution

In order to improve robustness, many statistical model-based methods have been proposed to constrain the evolution of level set functions. Tsai *et al.* [9] integrated parametric statistical shape modeling with the region-based curve evolution to

drive the curve by using the statistical shape model as constraints. The shape variability is estimated by performing PCA on the globally aligned level set functions, and the major shape variation is then statistically reflected by the changes along the principal components. Subsequently, a parametric level set function Φ can be described as a function of the feature vector in the PCA space, $\mathbf{w}$, and the pose parameter $\mathbf{p}$, reflecting a global transformation from the shape space onto the image space (please refer to Eq.(3) for details). Then, the associated energy functions (similar to Eq.(1)) can be minimized by finding the optimal parameters $\hat{\mathbf{w}}$ and $\hat{\mathbf{p}}$, where $\hat{\mathbf{w}}$ is constrained by the PCA model. The energy function can be defined as follows,

$$E_{img}(\mathbf{w}, \mathbf{p}, I) = {S_u}^2/A_u + {S_v}^2/A_v. \tag{2}$$

Notice that the parametric level set function Φ is determined by $\mathbf{w}$ and $\mathbf{p}$, and it divides the image domain into two regions — inside $R^u = \left\{\mathbf{x} \in \Re^2 : \Phi(\mathbf{x}) < 0\right\}$ and outside $R^v = \left\{\mathbf{x} \in \Re^2 : \Phi(\mathbf{x}) > 0\right\}$ the zero level set curve, and the areas of these two regions are $A_u = \int_\Omega H(-\Phi[\mathbf{w}, \mathbf{p}](\mathbf{x}))d\mathbf{x}$ and $A_v = \int_\Omega H(\Phi[\mathbf{w}, \mathbf{p}](\mathbf{x}))d\mathbf{x}$ ($H()$ is the regularized Heaviside function), and $S_u = \int_\Omega I \cdot H(-\Phi[\mathbf{w}, \mathbf{p}])d\mathbf{x}$ and $S_v = \int_\Omega I \cdot H(\Phi[\mathbf{w}, \mathbf{p}])d\mathbf{x}$ are the total intensities of the two regions.

2.3 Statistical Models of Level Set Functions

The PCA-based statistical model of the level set functions can be constructed according to [9]. First, all the N sample images are globally aligned onto the standard shape space, and the level set functions of these aligned images are $\{\Phi_i\}_{i=1,2,..,N}$. Then the PCA model is constructed using these N sample level set functions. Notice that each level set function Φ_i is digitalized and re-arranged into a 1D vector in order to perform PCA. For simplicity, we still use the same symbol for both the original and the re-arranged level set functions. According to PCA, a parametric level set function can be represented by,

$$\Phi[\mathbf{w},\mathbf{p}](\mathbf{x}) = \bar{\Phi}(\tilde{\mathbf{x}}) + \sum_{k=1}^{K} w_k \varphi_k(\tilde{\mathbf{x}}), \qquad \tilde{\mathbf{x}} = T(\mathbf{p})\mathbf{x}, \tag{3}$$

where $\tilde{\mathbf{x}}$ represents a point in the shape domain, and $\mathbf{x}$ is a point in the image domain. $T(\mathbf{p})$ is a global transformation matrix consisting of the translation vector $\mathbf{t}$, the rotation matrix R_θ and the scaling factor χ. $\bar{\Phi} = 1/N(\sum_{n=1}^{N} \Phi_i)$ is the mean level set function, and φ_k is the kth eigenvector of the covariance matrix of the N samples, and ψ_k is the corresponding eigenvalue. K largest eigenvalues (and corresponding eigenvectors) are chosen as the principal components to statistically represent the major variations of the level set functions.

We constructed two kinds of statistical models in experiments: one for putamen shapes and the other for the shapes of ventricle frontal horns. Altogether, 13 images are used for training the putamen shape models. Fig.1 illustrates five signed distance function(SDF)-based training samples of putamens, and Fig.2 (top row) gives the corresponding binary putamen shapes. These SDFs are used

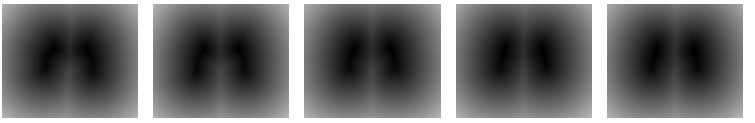


Fig. 1. Five training SDFs of 13 total samples for putamen shapes

Fig. 2. Five binary sample images for putamens *(top)* and ventricle shapes *(bottom)*

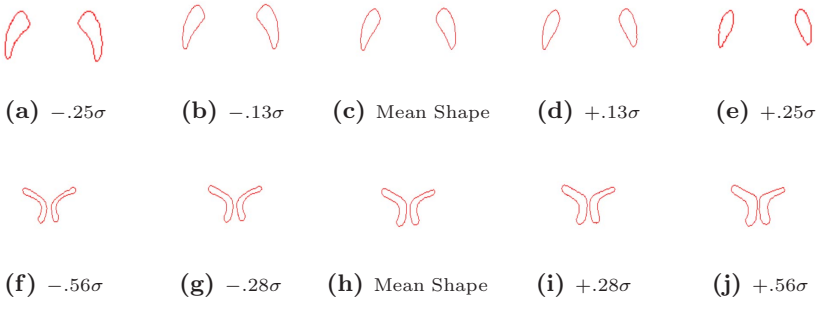


Fig. 3. Shapes by varying the weights for the first principal component ($\sigma = \sqrt{\psi_1}$)

as the training samples in the PCA model. Four principal components are finally chosen, which can represent about 97.5% of the total energy of the space spanned by the training set.

For constructing the statistical model of ventricles, we used 14 training images as the training data, and four principal components that fit 95.4% of the total energy have been chosen. Fig.2 (bottom row) illustrates five samples of the binary images of the ventricle shapes. Fig.3 illustrates the shapes reconstructed by varying the weights for the first principal component of the PCA model. It can be seen that the major shape changes have been captured by the first principal component.

3 Our Novel Joint Curve Evolution Algorithm

As stated in the introduction, non-parametric level set curve evolution can match image boundaries accurately but often suffer from local minima, and parametric

level set curve evolution is more robust by using statistical shape models but might be less accurate in matching object boundaries. In this work, we propose a new joint curve evolution algorithm by integrating the advantages of both parametric and non-parametric level set curve evolution methods.

3.1 Joint Curve Evolution Using MAP Framework

Based on the principle of MAP, the object shape S in an input image I can be estimated by,

$$\widehat{S} = \arg\max_{S}\{\log P(S|I)\}, \tag{4}$$

where $P(S|I)$ is the posteriori of the object shape. For parametric representation, a shape can be calculated from its feature vector $\mathbf{w}$ based on the statistical model described in Section 2.3, and we denote the parametric level set function or parametric shape as Φ or $\Phi[\mathbf{w}]$; for non-parametric representation, the non-parametric shape can be described by the level set function ϕ in the image domain. In this work, the object shape that is jointly represented by both ϕ and Φ (and hence $\mathbf{w}$), *i.e.*, $S = (\phi, \mathbf{w})$, and the task of segmentation can be formulated by,

$$(\widehat{\phi}, \widehat{\mathbf{w}}) = \arg\max_{\phi,\mathbf{w}}\{P(I|\phi, \mathbf{w})P(\phi|\mathbf{w})P(\mathbf{w})/P(I)\}, \tag{5}$$

where $P(I|\phi, \mathbf{w})$ is the probability of image I given (ϕ, $\mathbf{w}$). $P(\mathbf{w})$ determines the shape prior explicitly, and $P(\phi|\mathbf{w})$ is the conditional probability which acts as a bridge between the parametric and the non-parametric object shape representations. It is worth noting that the parametric and non-parametric level set functions, Φ and ϕ, represent different curves in the image domain. The former is calculated from the statistical model and is more constrained by the shape priors, and the latter can match the image boundaries better. The conditional probability makes sure that these two shapes are close enough so that the non-parametric shape ϕ (acts as the final result) is not only constrained by the statistical model via $\mathbf{w}$ but also matches object boundaries accurately.

Assuming these probabilities are subject to Gibbs distribution, the joint object shape can be obtained by minimizing the following energy function,

$$E_{total}(\phi, \mathbf{w}) = E_{img}(\mathbf{w}, \phi, I) + \lambda_p E_{prior}(\mathbf{w}) + \lambda_s E_{similarity}(\phi, \mathbf{w}), \tag{6}$$

where λ_p and λ_s are the weighting coefficients for the energy terms. The three energy terms are described in detail as follows.

The first energy term E_{img} reflects the matching degree between the joint object shape $(\mathbf{w}, \phi)$ and the image data, and it can be decoupled into two terms: $E_{img}(\mathbf{w}, \phi, I) = E_{img}(\phi, I) + E_{img}(\mathbf{w}, I)$. The first term reflects the matching degree between the non-parametric curve ϕ and the image I. The second term reflects the matching degree between the parametric curve $\Phi[\mathbf{w}]$ and the image I, and it drives the parametric curve toward the object in the image. Therefore,

the definition of the first term is similar the Eq.(1), and the second term is similar to Eq.(2) (by assuming no global transformation). It can be seen that the parametric shape $\Phi[\mathbf{w}]$ is constrained by the statistical model and is more robust in object matching, and the non-parametric shape ϕ can be used to match any arbitrary smooth object shapes and is more accurate in matching object boundaries.

The second energy term E_{prior} constrains the variations of the parametric shape $\Phi[\mathbf{w}]$ according to the prior distribution trained from sample shapes using PCA. Using the statistical model built in Section 2.3, E_{prior} can be defined as,

$$E_{prior}(\mathbf{w}) = \mathbf{w}^T \Psi^{-1} \mathbf{w}, \tag{7}$$

where Ψ is the diagonal matrix containing K largest eigenvalues, $\psi_k, k = 1, .., K$, according to the PCA model.

The third energy term $E_{similarity}$ reflects the difference between the parametric and non-parametric curves. Since the joint object shape S includes both the parametric and non-parametric level set functions, $\Phi[\mathbf{w}]$ and ϕ, it is necessary to constrain them so that they represent similar shapes,

$$E_{similarity}(\phi, \mathbf{w}) = \int_{\Omega} (H(\phi(\mathbf{x})) - H(\Phi[\mathbf{w}](\mathbf{x})))^2 d\mathbf{x}, \tag{8}$$

where Ω represents the image domain, and $H()$ is the regularized Heaviside function.

3.2 Curve Evolution with Pose Invariance

Section 3.1 does not consider the global transformations among the training samples. In fact all the training samples can be globally aligned into a standard shape domain before training the statistical model so that no variations of the position of images have been included in the statistical model. Therefore, in this subsection, by considering the pose parameter $\mathbf{p}$ between the standard shape domain and the image domain, a parametric curve Φ can be represented by $(\mathbf{w}, \mathbf{p})$, and thus we augment the proposed Bayesian formulation in Eq.(5) as,

$$\begin{aligned}(\widehat{\phi}, \widehat{\mathbf{w}}, \widehat{\mathbf{p}}) &= \arg\max_{\phi, \mathbf{w}, \mathbf{p}} \{P(I|\phi, \mathbf{w}, \mathbf{p}) P(\phi|\mathbf{w}, \mathbf{p}) P(\mathbf{w}) P(\mathbf{p}) / P(I)\} \\ &= \arg\max_{\phi, \mathbf{w}, \mathbf{p}} \{P(I|\phi, \mathbf{w}, \mathbf{p}) P(\phi|\mathbf{w}, \mathbf{p}) P(\mathbf{w})\},\end{aligned} \tag{9}$$

by assuming that the distributions of ϕ and $\mathbf{w}$ are independent with that of $\mathbf{p}$, and $\mathbf{p}$ is subject to uniform distribution, the associated energy function is defined as,

$$E_{total}(\phi, \mathbf{w}, \mathbf{p}) = E_{img}(\phi, \mathbf{w}, \mathbf{p}, I) + \lambda_p E_{prior}(\mathbf{w}) + \lambda_s E_{similarity}(\phi, \mathbf{w}, \mathbf{p}). \tag{10}$$

Notice that the image energy term $E_{img}(\phi, \mathbf{w}, \mathbf{p}, I)$ can also be decoupled into two terms,

$$E_{img}(\phi, \mathbf{w}, \mathbf{p}, I) = E_{img}(\phi, I) + E_{img}(\mathbf{w}, \mathbf{p}, I), \tag{11}$$

reflecting the matching degree between the non-parametric and parametric curves and the image. In our implementation, we use Eq.(1) and Eq.(2) to calculate the first term and the second term of Eq.(11), respectively.

3.3 Minimization of Energy Function

The energy function of Eq.(10) can be minimized by iteratively updating $\mathbf{w}$, $\mathbf{p}$ and ϕ using gradient descent method.

Updating w. First the parametric vector $\mathbf{w}$ can be updated by assuming that $\mathbf{p}$ and ϕ are known. The gradient of E_{total} with respect to $\mathbf{w}$ is,

$$\bigtriangledown_{\mathbf{w}} E_{total} = \bigtriangledown_{\mathbf{w}} E_{img} + \lambda_p \bigtriangledown_{\mathbf{w}} E_{prior} + \lambda_s \bigtriangledown_{\mathbf{w}} E_{similarity}. \tag{12}$$

Since non-parametric contour ϕ is not related to $\mathbf{w}$, based on Eq.(11), only Eq.(2) is used to calculate the first term of Eq.(12). According to Eq.(7) and Eq.(8), Eq.(12) can be calculated as,

$$\begin{aligned}\bigtriangledown_{\mathbf{w}} E_{total} = &\left[-2(\frac{S_u}{A_u} \bigtriangledown_{\mathbf{w}} S_u + \frac{S_v}{A_v} \bigtriangledown_{\mathbf{w}} S_v) + (\frac{S_u}{A_u})^2 \bigtriangledown_{\mathbf{w}} A_u + (\frac{S_v}{A_v})^2 \bigtriangledown_{\mathbf{w}} A_v\right] \\ &+ \left[\Psi^{-1}\mathbf{w}\right] + \left[2 \oint_{\boldsymbol{C}} \Big(H(\Phi[\mathbf{w},\mathbf{p}] - H(\phi)) \bigtriangledown_{\mathbf{w}} \Phi[\mathbf{w},\mathbf{p}]\Big) ds\right]. \end{aligned} \tag{13}$$

Notice that the last term is an integral along the zero level set curve $\boldsymbol{C}$ of Φ, S_u/A_u is the average intensity in region R^u and S_v/A_v is the average intensity in region R^v. Term $\bigtriangledown_{\mathbf{w}} E_{total}$ is interpreted as the parametric curve evolution of the level set function.

Updating p. In the second step, the pose information $\mathbf{p}$ is updated by assuming that $\mathbf{w}$ and ϕ are known and fixed, and the gradient of E_{total} with respect to $\mathbf{p}$ can be calculated as,

$$\bigtriangledown_{\mathbf{p}} E_{total} = \bigtriangledown_{\mathbf{p}} E_{img} + \lambda_s \bigtriangledown_{\mathbf{p}} E_{similarity}, \tag{14}$$

where

$$\bigtriangledown_{\mathbf{p}} E_{img} = (\frac{S_u}{A_u})^2 \bigtriangledown_{\mathbf{p}} A_u + (\frac{S_v}{A_v})^2 \bigtriangledown_{\mathbf{p}} A_v - 2\frac{S_u}{A_u} \bigtriangledown_{\mathbf{p}} S_u - 2\frac{S_v}{A_v} \bigtriangledown_{\mathbf{p}} S_v, \tag{15}$$

and S_u, S_v, A_v, A_u are described in section 2.2 as the functions of $\Phi[\mathbf{w},\mathbf{p}](\mathbf{x})$. The second term of Eq.(14) can be calculated as,

$$\bigtriangledown_{\mathbf{p}} E_{similarity} = 2 \oint_{\boldsymbol{C}} \Big(H(\Phi[\mathbf{w},\mathbf{p}] - H(\phi)) \bigtriangledown_{\mathbf{p}} \Phi[\mathbf{w},\mathbf{p}]\Big) ds. \tag{16}$$

Notice that this line integral is calculated along curve $\boldsymbol{C}$ of Φ and $\bigtriangledown_{\mathbf{p}} \Phi[\mathbf{w},\mathbf{p}](\mathbf{x}) = \bigtriangledown_{\tilde{\mathbf{x}}} \Phi[\mathbf{w},\mathbf{p}] \cdot (\frac{\partial T}{\partial \mathbf{p}}\mathbf{x})$.

Updating ϕ. The third step updates the non-parametric level set function ϕ by fixing $\mathbf{w}$ and $\mathbf{p}$. The gradient of the total energy with respect to ϕ is,

$$\bigtriangledown_\phi E_{total} = \bigtriangledown_\phi E_{img} + \lambda_s \bigtriangledown_\phi E_{similarity}. \tag{17}$$

According to Eq.(11), only the non-parametric part of the energy function $E_{img}(\phi, I)$ is used to calculate the first term above. Therefore, the gradient of $\bigtriangledown_\phi E_{total}$ with respect to ϕ is calculated as

$$\begin{aligned}\bigtriangledown_\phi E_{total} = \delta_\epsilon(\phi)\Big(\nu div(\frac{\bigtriangledown\phi}{|\bigtriangledown\phi|}) - (I(\mathbf{x}) - c_W)^2 + (I(\mathbf{x}) - c_{\setminus W})^2\Big) \\ +\lambda_s\Big(2\oint_{\boldsymbol{C}}(H(\phi) - H(\Phi[\mathbf{w}, \mathbf{p}]))ds\Big),\end{aligned} \tag{18}$$

where the last term is the integral calculated along the zero level set of Φ, *i.e.*, curve C. Notice that $\bigtriangledown_\phi E_{total}$ represents non-parametric curve evolution of the joint curve evolution.

The above three steps are iteratively performed in order to update the parametric curve representation $\Phi[\mathbf{w}, \mathbf{p}]$ as well as the non-parametric curve representation ϕ until convergence.

4 Results

In order to evaluate the performance of the proposed joint curve evolution algorithm, in this section, we applied the algorithm to extract the ventricle frontal horns and putamens from MR brain images. The reason that we choose these two structures is that the boundaries of the ventricle frontal horn are very clear, while the white matter/gray matter contrast of putamen boundaries is lower compared to other anatomical structures of the brain. Thus we can fully evaluate the performance of the algorithm under different conditions. The MR brain images used in the experiments are obtained from the MRIcro website [17], and prior to the experiments, corresponding slices from different images are selected and the ventricle frontal horn and putamen shapes are manually marked to act as both ground truth and training samples of the statistical models. Altogether, we used 14 ventricle images and 13 putamen slices in the experiments, and the pixel spacing of the images is $0.9375mm \times 0.9375mm$. Leave-one-out strategy is used for training and testing the algorithms. In each iteration of the cross-validation, the MR brain image of one subject is left out from the training data that are used for training the statistical model, and is tested by using the joint curve evolution algorithm trained on them. The validation iterates until all the images are left out once and only once for testing. The average distance between the resultant curve and the ground truth is then recorded, reflecting the accuracy of the segmentation algorithm with respected to that testing image.

Fig. 4 illustrates the results for matching the ventricle frontal horns using our methods on 14 test images, and Fig. 5 gives the matching results for putamens. The initialization is done as follows. We manually indicate the location of the

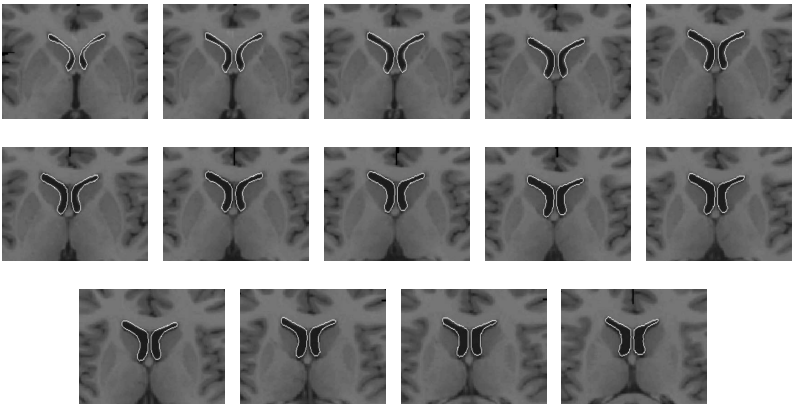


Fig. 4. All the 14 segmentation results for the ventricle frontal horns of MR images by using the joint curve evolution algorithm with leave-one-out training strategy

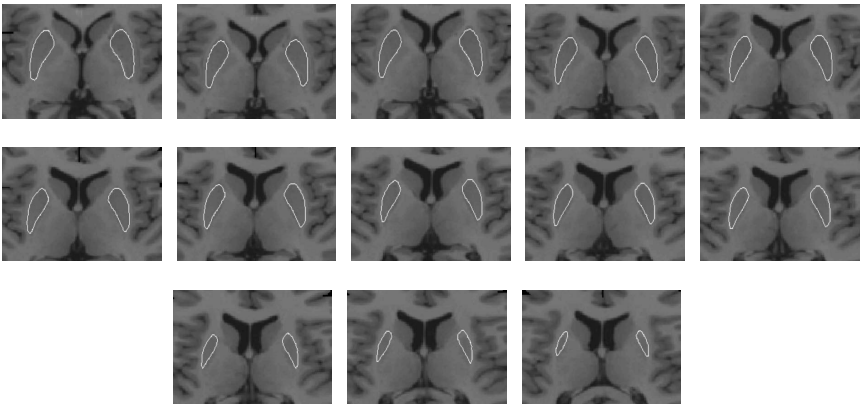


Fig. 5. All the 13 segmentation results of putamens by using our proposed algorithm with leave-one-out training strategy

objects (*i.e.*, the visual estimate of the object center), and then rigidly shift the mean shape to that location, acting as the initial curve. We performed more than 3 initializations for each image, and the errors for manual estimation of the object center are within the range of [-7mm, 7mm], and the proposed algorithm obtained satisfactory results for all the tests with different initializations. The results in Fig. 4 and Fig. 5 also indicate that the proposed algorithm is not only effective for shapes with highly distinct boundaries such as the ventricles, but also satisfactory with low contrast shapes such as the putamens. We also evaluated the object matching performance by quantitatively comparing the results of the proposed joint curve evolution and the parametric-only curve evolution (Tsai method [9]) with the manual ground truth. For comparison purposes, the same leave-one-out strategy was also used for Tsai method.

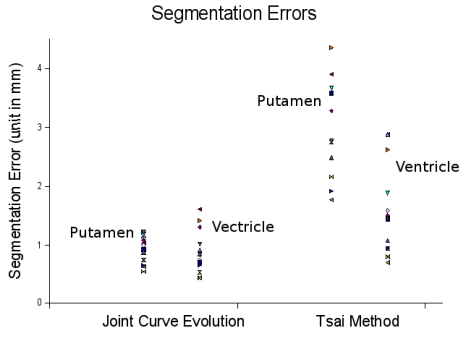


Fig. 6. Comparison results for extracting the shapes of the ventricle frontal horn and putamens corresponding to Fig.4 and Fig.5, respectively

Fig. 6 plots the segmentation errors for matching both putamens and ventricles in all the testing images. It can be seen that the matching errors of the proposed joint evolution algorithm are much lower than those of Tsai method. For putamen, the average and standard deviation of our algorithm and Tsai method are 0.95mm/0.24mm and 3.06mm/0.81mm, respectively; for ventricle segmentation, they are 0.89mm/0.37mm and 1.59mm/0.76mm, respectively. Obviously we obtained better segmentation accuracy for ventricles than putamens, which is because that the boundaries of ventricles are much more clearer. In summary, both visual and quantitative results indicate that the proposed joint curve evolution algorithm is not only robust but also more accurate than the parametric curve evolution method for medical image segmentation.

5 Conclusion

We have proposed a joint curve evolution algorithm for medical image segmentation. In this algorithm, both parametric and non-parametric curve evolution methods are employed based on the Bayesian framework to extract object shapes using level set functions. As a result, the algorithm has the advantages of both parametric and non-parametric curve evolution. It is as robust as the statistical model-based parametric curve evolution algorithm and at the same time, yields more accurate segmentation results. In the future work, we will extend our algorithm to match various anatomical structures in 3D medical images and improve the algorithm so that it can be used to segment multiple objects simultaneously.

References

1. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models-their training and application. Computer Vision and Image Understanding 61(1), 38–59 (1995)
2. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(6), 681–685 (2001)

3. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models, pp. 259–268. IEEE Comput. Soc. Press, Los Alamitos (1987)
4. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. International Journal of Computer Vision 22(1), 694–699 (1997)
5. Malladi, R., Sethian, J.A., Vemuri, B.C.: Shape modeling with front propagation: a level set approach. Transactions on Pattern Analysis and Machine Intelligence 17(2), 158–175 (1995)
6. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed - algorithms based on hamilton-jacobi formulations. Journal of Computational Physics 79(1), 12–49 (1988)
7. Leventon, M.E., Grimson, W.E.L., Faugeras, O.: Statistical shape influence in geodesic active contours. In: 5th IEEE EMBS International Summer School on Biomedical Imaging, 2002 (2002)
8. Chan, T.F., Vese, L.A.: Active contours without edges. IEEE Transactions on Image Processing 10(2), 266–277 (2001)
9. Tsai, A., Yezzi, A., Wells, W., Tempany, C., Tucker, D., Fan, A., Grimson, W.E., Willsky, A.: A shape-based approach to the segmentation of medical imagery using level sets. IEEE Transactions on Medical Imaging 22(2), 137–154 (2003)
10. Chen, Y.M.: Using prior shapes in geometric active contours in a variational framework. International Journal of Cmputer Vision 50(3), 315–328 (2002)
11. Rousson, M., Paragios, N.: Prior knowledge, level set representations & visual grouping. International Journal of Computer Vision 76(3), 231–243 (2008)
12. Cremers, D., Kohlberger, T., Schnrr, C.: Nonlinear shape statistics in mumfordshah based segmentation. In: Computer Vision ECCV, pp. 516–518 (2002)
13. Cremers, D., Osher, S., Soatto, S.: Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. International Journal of Computer Vision 69(3), 335–351 (2006)
14. Kim, J., Cetin, M., Willsky, A.: Nonparametric shape priors for active contour-based image segmentation. Signal Processing 87(12), 3021–3044 (2007)
15. Rousson, M., Paragios, N., Deriche, R.: Implicit active shape models for 3d segmentation in mr imaging. In: Barillot, C., Haynor, D.R., Hellier, P. (eds.) MICCAI 2004. LNCS, vol. 3216, pp. 209–216. Springer, Heidelberg (2004)
16. Bresson, X., Vandergheynst, P., Thiran, J.P.: A variational model for object segmentation using boundary information and shape prior driven by the mumford-shah functional. International Journal of Computer Vision 68(2), 145–162 (2006)
17. Rorden, C.: Mricro (2005), `http://www.sph.sc.edu/comd/rorden/mricro.html`

Localizing Objects with Smart Dictionaries

Brian Fulkerson, Andrea Vedaldi, and Stefano Soatto

Department of Computer Science,
University of California, Los Angeles, CA 90095
{bfulkers,vedaldi,soatto}@cs.ucla.edu

Abstract. We present an approach to determine the category and location of objects in images. It performs very fast categorization of each pixel in an image, a brute-force approach made feasible by three key developments: First, our method reduces the size of a large generic dictionary (on the order of ten thousand words) to the low hundreds while increasing classification performance compared to k-means. This is achieved by creating a discriminative dictionary tailored to the task by following the information bottleneck principle. Second, we perform feature-based categorization efficiently on a dense grid by extending the concept of integral images to the computation of local histograms. Third, we compute SIFT descriptors densely in linear time. We compare our method to the state of the art and find that it excels in accuracy and simplicity, performing better while assuming less.

1 Introduction

Bag-of-features methods have enjoyed great popularity in object categorization, owing their success to their simplicity and to surprisingly good performance compared to more sophisticated models and algorithms. Unfortunately, such methods only provide an answer as to whether an image contains an object of a certain category, but they do not offer much insight as to where that object might be within the image. In other words, because the representation discards spatial information, bag-of-features methods cannot be used for localization directly.

That is, unless one could devise an object categorization method efficient enough to test at a window centered on each pixel of an image. In that case, one would be able to exploit the co-occurrence of features within a local region and localize the object, pixel by pixel. However, with many features detected in each image and quantized into thousands or tens of thousands of "words," this does not appear to be a viable proposition, especially in light of recent results that advocate using very large visual dictionaries [1,2,3].

But what if we could reduce the size of a dictionary from tens of thousands of words to a few hundred and maintain improved localization? After all, dictionaries commonly used in bag-of-features are not designed for the specific task of categorization, so there may be gains to be found in creating "smarter" dictionaries that are tailored to the task. This is precisely what we set out to do. With this we can obtain robust, efficient localization, and show that our scheme

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 179–192, 2008.

Fig. 1. Upper Left: Original image. Middle: Labeling weighted by the confidence for the class "person". Lower Left: Labeling weighted by the confidence, with low confidence background pixels reclassified as foreground. Right: Labeling weighted by the confidence, with low confidence foreground pixels reclassified as background.

performs better than the state of the art [4] on a challenging dataset [5] despite its simplicity.

Contributions. In this paper we propose a method for pixel-level category recognition and localization. We employ a simple representation (bag-of-features) and contribute three techniques which make the categorization process efficient. First, we extend integral images [6] to windowed histogram-based classification. Second, we construct small dictionaries which maintain the performance of their larger counterparts by using agglomerative information bottleneck (AIB) [7]. In order to greatly reduce the bottleneck of quantizing features, we construct the large dictionaries using hierarchical k-means (HKM). We also propose an important speedup which makes it possible to compute AIB on large dictionaries with ease. Third, we show that we can compute SIFT features densely in linear time.

Related work. Lazebnik *et al.* [8] also perform discriminative learning to optimize k-means, but are limited to small dictionaries and visual words which are Voronoi cells. Leibe *et al.* [9] also perform compression, but not in a discriminative sense. Finally, Winn *et al.* [10] do discriminative compression in a similar fashion, but we show that we perform better and can scale to larger dictionaries. For the task of pixel-level localization, we show that our method outperforms k-means and Winn *et al.*, while being nearly two hundred times faster to construct. We compare our method directly to Winn *et al.* [10] in Sect. 3.1.

Object categorization methods have matured greatly in recent years, going beyond bags of features by incorporating a spatial component into their model. Approaches are varied, but broadly tend to include one of the following: interactions between pairs of features [11,12,13], absolute position of the features [14], segmentation [15,16], or a learned shape or parts model of the objects [4,17,18]. Our method exploits interaction between groups of features (all features in the

window), but does not explicitly represent their configuration, in the spirit of achieving viewpoint-invariance for objects of general shape[19].

Regarding object localization, recent works are based on two different approaches: either they form a shape based model of the object class as in [4,17], or they enforce spatial consistency using a conditional random field (CRF) [20,21]. We focus our comparisons on the method of Marszalek *et al*. [4], who forms a family of shape models for each category from the training data and casts these into the target image on sparse feature points to find local features which agree on the deformation of one of the learned models. Our approach will show that we obtain better performance by simply performing local classification at every pixel.

Along the way, we will construct a small, smart dictionary which is comprised of clusters of features from a much larger dictionary using AIB [7]. Liu *et al*. [22] recently proposed a co-clustering scheme maximizing mutual information (MMI) for scene recognition. Agarwal *et al*. [23] cluster features to create a whole image descriptor called a "hyperfeature" stack. Their scheme repeatedly quantizes the data in a fixed pyramid, while our representation allows the computation of any arbitrary window without incurring any additional computational penalty. We can just as easily extract our bag-of-features for the whole image, blocks of the image, or (as we show) each pixel on a grid.

After this paper was submitted, two additional related works were published. Lampert *et al*. [24] use branch-and-bound to search all possible subwindows of an image for the window which best localizes an object. They do not seek to localize at the pixel level, or handle multiple objects in one image. Shotton *et al*. [25] perform pixel labeling as we do, but use much simpler features combined with randomized decision forests. Because they use simple features, they must build their viewpoint invariance by synthetically warping the training images and providing them as new training examples. Our framework allows for that, but our descriptors already exhibit reduced sensitivity to viewpoint.

2 Brute-Force Localization

Our method uses bag-of-features as a "black box" to perform pixel-level category recognition. In the simplest case, this involves extracting features from the image and then for each window aggregating them into a histogram and comparing this histogram with the training data we have been provided. The main stumbling block is the extraction of histograms at each pixel of the image. For this, we use integral images. However, this alone is not sufficient: Using large dictionaries in the setting we propose would be impossible, yet we need our dictionary to remain discriminative in order to be useful. To this end, in Sect. 3 we propose a method for building a compact, efficient and informative dictionary from a much larger one.

Integral Images. Viola *et al*. [6] popularized the use of integral images for the task of feature extraction in boosting, and it has since been used by others [20]

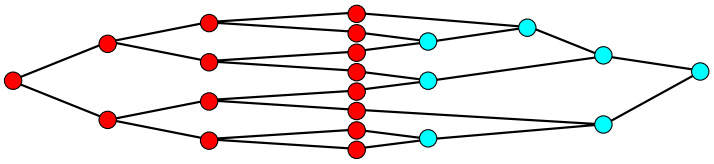


Fig. 2. Dictionary architecture. We use hierarchical k-means (HKM) to build a vocabulary tree (left, red nodes) of finely quantized features by recursively partitioning the data. Next, we use AIB to build an agglomerative tree (right, blue nodes) of informative words. This architecture is efficient (in training and testing) and powerful.

for similar purposes. Integral images can also be used to quickly count events in image regions [26], and Porikli [27] shows how to compute integral histograms in Cartesian spaces. We build integral images of spatial occurrences of features and use them to efficiently extract histograms of visual words on arbitrary portions of the image. For each visual word b in our dictionary, let $O_b(x, y)$ be the number of occurrences of b at pixel (x, y) (typically this number is either zero or one). Each image O_b is transformed into a corresponding integral image I_b by summing over all the pixels $(x', y') \leq (x, y)$ above and to the left of pixel (x, y):

$$I_b(x, y) = \sum_{x'<x} \sum_{y'<y} O_b(x', y')$$

Let R be a rectangular image region. The histogram $H_R(b)$ is the number of occurrences of b in R and can be quickly computed as:

$$H_R(b) = I_b(x_s, y_s) + I_b(x_e, y_e) - I_b(x_s, y_e) - I_b(x_e, y_s)$$

where (x_s, y_s) is the upper left corner and (x_e, y_e) is the lower right corner of R. In this way we can extract a histogram of feature occurrences for a window of arbitrary size in constant time. The memory required scales with the size of the image and the size of the dictionary, and the constant time required to construct each histogram scales linearly with the size of the dictionary. This precludes the use of very large dictionaries, because each dictionary element that is included requires adding an integral image.

3 Informative, Compact and Efficient Dictionaries

Our localization method directly benefits from having a small dictionary because the complexity is linear in its size. Yet many recent works [1,2,3,29] indicate that large or very large dictionaries perform better for both object recognition and categorization. However, over-specific visual words should eventually over-fit the data, especially in categorization. We argue that one of the reasons why large dictionaries often outperform smaller ones is that dictionaries are usually not optimized for discrimination. If visual words could be tailored to discriminate

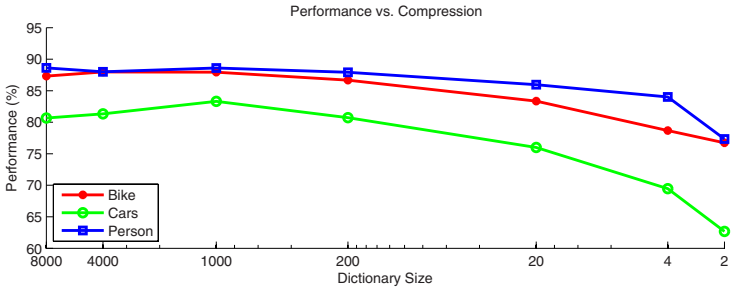


Fig. 3. Results of an experiment showing the performance of AIB as the dictionary is compressed. We adopt the framework of [28] on Graz-02, extracting SIFT descriptors on salient regions, quantizing them, and classifying the resulting histograms with an SVM. We vary the compression of the dictionary, starting from the full HKM tree (8,000 leaves, K=20) and compressing to a dictionary with only 2 elements. In each case, we can compress the dictionary by a factor of 8 without losing any accuracy. In some cases (Cars, Bikes) we even increase performance slightly.

different categories, a smaller number of them would be sufficient. Motivated by this idea, we seek to gain the performance increases of recent approaches using large dictionaries without their computational burden.

Winn *et al*. [10] introduced the idea of constructing small and informative visual dictionaries by compressing larger ones. Here we propose a novel architecture and compression algorithm that has two key advantages: (i) it is very fast to project novel features on the optimized dictionary and (ii) compression is several orders of magnitude faster, which makes it possible to operate on much larger dictionaries and datasets. In addition, we show that our method outperforms [10] for the task of pixel level categorization (Sect. 4).

Fast Projection by HKM. In order to project N novel features $f \in \mathcal{F} \subset \mathbb{R}^n$ onto a visual dictionary of L elements, the required time is usually $O(NL)$. This is true even if the dictionary is eventually compressed into a smaller one [10]. Since a large number of features N are typically extracted from an image, mapping features to the visual dictionary may become the bottleneck of the recognition pipeline.

Here we solve this problem by using an HKM [1] tree as the initial visual dictionary. HKM trees have shown excellent performance in object recognition [1,2]. More importantly, they enable efficient projection of novel features, requiring only $O(N \log L)$ operations. Combining the HKM tree with the compression tree (Sect. 3.1), yields the coarse-to-fine-to-coarse architecture of Fig. 2.

3.1 Dictionary Compression

We compress a visual dictionary by merging visual words in such a way that the discriminative power of the dictionary is preserved. The discriminative power can be characterized in different ways, yielding different compression algorithms. Here we

discuss and compare two: Agglomerative Information Bottleneck (AIB) [7] and the method from [10], which we indicate with WCM. We also contribute a modification of the AIB algorithm that makes it feasible to process dictionaries of tens of thousands of elements. We show that the same fast algorithm may be used to speed-up WCM as well. However, even with this speedup we find that WCM is much slower than AIB (to the point of being infeasible for large datasets and dictionaries) and performs worse than AIB when applied to pixel-level categorization.

AIB Compression. AIB characterizes the discriminative power of the dictionary $\mathcal{X}$ as the mutual information $I(x, c)$ of the random variables x (visual word) and c (category):

$$I(x,c) = \sum_{x \in X} \sum_{c=1}^{C} P(x,c) \log \frac{P(x,c)}{P(x)P(c)}. \tag{1}$$

The joint probability $P(x, c)$ is estimated from data simply by counting the number of occurrences of each visual word $x \in \mathcal{X}$ in each category $c \in \{1, \dots, C\}$. AIB iteratively compresses the dictionary $\mathcal{X}$ by merging the two visual words x_i and x_j that cause the smallest decrease D_{ij} in the mutual information (discriminative power) $I(x, c)$. Denoting $[x]_{ij}$ the random variable corresponding to the dictionary after the merge, the quantity D_{ij} is

$$D_{ij} = I(x,c) - I([x]_{ij}, c). \tag{2}$$

The information $I(x, c)$ is monotonically reduced after each merge. Merging is iterated until one obtains the desired number of words.

At test time, projecting a visual word $x \in \mathcal{X}$ onto the compressed dictionary requires constant time ($O(1)$). So, since we use HKM for the initial dictionary, the number of operations required to project N novel features on the compressed dictionary is only $O(N \log K)$, where K is the number of leaves of the HKM tree.

In Fig. 3 we show the effectiveness of this technique using a simple experiment on Graz-02. In all cases, we compress the dictionary significantly without losing any accuracy. In fact, in two of the three cases the results are slightly improved at some compression level.

Fast AIB. The basic implementation of the AIB algorithm is prohibitively slow for very large dictionaries. The implementation proposed in Slonim *et al.* [7] stores the symmetric "distance" matrix $D = [D_{ij}]$ ($O(L^2)$ space).[1]

Then, at each iteration one only needs to update the row and column i, j of D which were involved in the last merge (since only words x_i and x_j change).

[1] **Reciprocal Nearest Neighbor Clustering** [9] proposes an efficient agglomerative clustering algorithm that can be applied whenever the distance matrix D_{ij} satisfies the *reducibility property* $D_{ij} \leq \min\{D_{ik}, D_{jk}\} \Rightarrow \min\{D_{ik}, D_{jk}\} \leq D_{\bar{ij},k}$, where $\bar{ij}$ denotes the merged dictionary entry. Unfortunately, AIB clustering violates this property. For a counter example, consider the case $C = 3$, $P(x_i) = P(x_j) = P(x_k) = 1/3$, $P(c = 1|x_k) = P(c = 2|x_k) = 1/3$, $P(c = 1|x_i) = P(c = 2|x_i) = 2/5$ and $P(c = 2|x_j) = P(c = 3|x_j) = 2/5$.

This has complexity $O(LC)$. Searching for the minimal matrix element at each step is $O(L^2)$, and this process is iterated L times, so the overall complexity is $O(L(L^2 + LC))$ time and $O(L^2)$ space [30].

A simple modification of the basic algorithm is far more efficient. We cache for each i the index and value (k_i, D_{ik_i}) of the minimum distance along the row and do not store D. This reduces the time spent searching for the minimum element (i^*, j^*) of D from $O(L^2)$ to $O(L)$. Now, when we merge (i^*, j^*), we must update the entries (k_i, D_{ik_i}) for which either $k_i = i^*$ or $k_i = j^*$. This has time complexity $O(L(L+\gamma LC))$, where γ is the number of entries which need to be updated at each iteration. We find empirically that $\gamma \ll L$, so in practice the amount of time taken is approximately $O(L^2C)$ and the space complexity has been reduced to $O(L)$.

To get a sense of the advantages of this implementation, the original AIB algorithm [30] requires L^2 elements of memory at each iteration, which meant that a 20,000 cluster case would require roughly 3.2GB of memory as opposed to 320kB with our modified approach. We also note that in the 10,000 cluster cases we test, we often find γ to be on the order of 5 and so the clustering process is very fast (about 5 minutes for 10,000 clusters on a 2.3Ghz Core 2 Duo). The basic implementation of AIB on the same task requires approximately a day.

WCM compression. WCM differs from AIB in the way it measures the discriminative power of the visual dictionary. This is motivated by the fact that in the bag-of-features setting images are represented by histograms of visual words rather than visual words in isolation. Thus, one is more interested in obtaining *informative histograms* than informative visual words. This notion could be captured, for instance, by considering the mutual information $I(h, c)$ in place of the information $I(x, c)$ used by AIB.

Due to the high dimensionality of the histograms, estimating $I(h, c)$ is nearly impossible without strong assumptions. WCM assumes that histograms are distributed according to a mixture of Gaussians, with one Gaussian per category. Moreover, they characterize the discriminative power of the dictionary by the category posterior probability $p(c|h)$ rather than by the information $I(h, c)$. This creates a mechanism for model selection which can automatically stop the merging procedure when a maximum of $p(c|h)$ is attained (in contrast, in AIB the information criterion $I(x, c)$ decreases monotonically). Finally, it is also possible to extend the fast AIB algorithm introduced in the previous section to WCM almost without changes.

Despite these appealing characteristics, WCM does not perform as well as AIB in our setting. First, despite our fast implementation, it is much slower than AIB on large datasets (in Sect. 4 we show it requires up to twelve days on a task that our fast AIB can solve in about five minutes).[2] Second, WCM model selection is not useful for our localization task as we are interested in obtaining

[2] Updating an entry of the D_{ij} matrix requires scanning the data to compute the linear correlation of bin i and j. This is due to the fact that WCM considers visual words in the context of histograms where AIB does not. Although the model assumes that histogram bins are statistically independent, they interact when merged. The update operation requires about $O(ML^2C)$, where M is the number of training histograms, as opposed to $O(L^2C)$ for AIB.

dictionaries of a prescribed size (Sect. 4). Third, AIB compressed dictionaries result in better categorization results than WCM[3] (Sect. 4; Table 1).

4 Experiments

Graz-02 [5] is a challenging dataset consisting of three categories (cars, bicycles, and people) with extreme variability in pose, scale and lighting. Our goal is the same as Marszalek *et al*. [4]: We wish to label each image pixel as either belonging to one of these categories or not. In order to compare directly to Marszalek *et al*. [4], we adopt their measure of performance: pixel precision-recall. Our features extraction and dictionary compression are implemented within VLFeat [31], and the rest of our implementation is available from our website[4].

Training. We select the same training images as [4], namely the first 150 odd numbered images from each category. We compute dense SIFT descriptors and quantize them using our dictionary (see Sect. 4). Then for each image we generate two histograms: The first aggregates all the features that belong to the background (based on the feature center and the ground truth object masks), and the second the features that belong to the object. This collection of histograms is used as training data for either an SVM classifier with χ^2 kernel or an inverse document frequency (IDF) [1] weighed k-nearest neighbor (KNN) classifier ($k = 10$).

Fast Dense Feature Extraction. We extract a SIFT descriptor [32] every four pixels. The support of each descriptor is a 16×16 patch. We do not compute the orientation of the descriptor since this has been shown to adversely affect other dense bag of features methods [28]. Features that have low gradient magnitude before normalization are discarded as in [14,3].

We introduce here a novel technique to compute dense SIFT descriptors very efficiently. Fast SIFT-like descriptors have been proposed by [33,3] and recently [34]. Our technique has the advantage of being fully equivalent to SIFT and still efficient: The complexity is only $O(Q^2 R)$ compared to $O(Q^2 R^2)$ of a direct implementation, where Q^2 the area of the image and R^2 the area of the descriptor support. Moreover, up to a small approximation, we can reduce the complexity to $O(Q^2)$, which is independent of the area of the descriptor support. Our implementation is included with VLFeat [31], an open source feature extraction library.

The idea is to reduce the calculation of the dense descriptors to a number of separable convolutions. Recall that the SIFT descriptor at location (x_0, y_0) is a three-dimensional histogram of the gradient $\nabla I(x, y)$ in a circular patch surrounding that point [32]. The histogram is indexed by the relative position $(x - x_0, y - y_0)$ and orientation $\angle \nabla I(x, y)$ of the gradient $\nabla I(x, y)$ in the patch, weighed by the gradient modulus $|\nabla I(x, y)|$ and by a Gaussian window

[3] This is probably due to the fact that in our setting the assumptions made by WCM are not satisfied.

[4] http://vision.ucla.edu/bag/

centered at (x_0, y_0). The relative positions are quantized in 4×4 bins and the orientation in 8 bins using bilinear interpolation. For a given orientation, the data for a bin b is obtained by computing integrals of the type $\int g(x - x_0, y - y_0)h_b(x - x_0, y - y_0)f(x, y)\,dx\,dy$, where $f(x, y)$ is the mass of the gradient at that particular orientation, $g(x, y)$ is the Gaussian window and $h_b(x, y)$ is the product of two triangular windows resulting from the bilinear interpolation of bin b. Since both $h(x, y)$ and $g(x, y)$ are separable, the calculation requires only $O(Q^2R)$ operations.

Notice that this requires $4 \times 4 \times 8$ separable convolutions in total. However, by dropping the Gaussian window $g(x, y)$ (the effect on the computed descriptors is modest), convolutions for different spatial bins at the same orientations are identical up to translation, and only 8 separable convolutions are sufficient. Moreover, recall that convolving by a rectangular kernel can be done very efficiently by integral images. Since convolving by a triangular kernel can be decomposed in convolving twice by rectangular ones, we obtain a final complexity of $O(Q^2)$.

We also experimented with color descriptors by first transforming the (R, G, B) image into the normalized (r, g, b) space [35] where $r = \frac{R}{R+G+B}$, $g = \frac{G}{R+G+B}$, $b = \frac{B}{R+G+B}$. SIFT descriptors are extracted independently from the r and g channel and concatenated into one 256 dimensional descriptor[5].

Dictionary Construction. We sample a large number of feature-category pairs from our training data and follow one of two approaches to construct a dictionary. As a baseline, we use k-means with $k = \{5, 40, 200\}$. Alternatively, we construct a hierarchical k-means dictionary with $k = 10$ and 10,000 leaf nodes, and then compress this dictionary to $N = \{5, 40, 200\}$ clusters (we experiment with both AIB and WCM). Notice that, in our application, the size of the dictionary is the primary factor in determining the speed and memory footprint of the classification algorithm.

Testing. We test on the first 150 even numbered images from each category. For each pixel on a gird with a step 4 pixels, we construct a histogram of feature occurrences within a window of 80×80 pixels using integral images (Sect. 2) and classify using either SVM or KNN. The classification returns a label and a score. The magnitude of the score indicates the confidence in the label and the sign of the score indicates the class (-1 is a fully confident classification of "background"). For pixels which do not lie on the grid, we interpolate the score from adjacent pixels.

We choose a range of confidence thresholds ρ and for each we classify as object all pixels which have a score greater than the threshold. These are compared to the ground-truth segmentation which provides us with pixel precision and recall for the testing data. We also use this threshold to create Fig. 1 and to generate the movie included as supplementary material.

[5] We do not include the b channel because the constraint $r + g + b = 1$ makes it redundant.

Table 1. A comparison of the pixel precision-recall equal error rates on Graz-02. Although we do not represent shape explicitly, our results are competitive with [4]. The best performance is achieved using our compressed dictionary (Sect. 3). We also outperform Winn *et al*. (WCM), and while our dictionaries take roughly 5 minutes to construct, Winn *et al*. takes up to 12 days on this task. Here time is the amount of time required per image, including dense feature extraction, quantization, and classification of all pixels. Dense feature extraction alone requires 0.15s for grayscale and 0.3s for RGB. Images are 640×480.

object class	cars	people	bicycles	time
[4] no hyp. eval.	40.4%	28.4%	46.6%	-
[4] no evid. collect.	50.3%	40.3%	48.9%	-
[4] full framework	53.8%	44.1%	61.8%	-
AIB5-KNN	39.8%	47.1%	57.4%	0.5s
AIB5-SVM	38.5%	48.2%	56.8%	0.7s
KM5-KNN	27.1%	32.1%	44.9%	2s
KM5-SVM	30.0%	33.1%	44.9%	2s
AIB40-KNN	47.5%	47.2%	61.7%	0.5s
AIB40-SVM	44.9%	49.0%	59.9%	0.8s
KM40-KNN	45.1%	42.8%	59.5%	2s
KM40-SVM	37.8%	45.4%	59.5%	2.5s
AIB200-KNN	50.9%	49.7%	63.8%	1.1s
AIB200-SVM	40.1%	50.7%	59.9%	3.3s
KM200-KNN	50.1%	46.5%	62.6%	2.5s
KM200-SVM	39.3%	49.3%	58.9%	5s
AIB200RGB-KNN	**54.7%**	47.1%	**66.4%**	1.4s
AIB200RGB-SVM	49.4%	**51.4%**	65.2%	3.7s
WCM200RGB-KNN	54.2%	41.1%	59.6%	1.4s
WCM200RGB-SVM	39.8%	46.3%	59.6%	3.7s
KM200RGB-KNN	51.6%	44.2%	60.8%	3.5s
KM200RGB-SVM	48.3%	49.3%	61.4%	7s

Discussion. Table 1 reports the points where precision and recall are equal and compares our results to those of Marszalek *et al*. [4], the previous state of the art in pixel accurate localization on Graz-02. The full curves are available at the author's website. Although we do not have shape or even scale in our model, we still perform significantly better on all categories. Specifically, our best performing cases are 4.8% better on bikes, 0.9% better on cars, and 7.3% better on people. In each case, the compressed dictionary outperforms the k-means dictionary of equal size. The differences decrease as the final vocabulary size is increased, which is intuitive because the variability of the dataset can be better captured by k-means as we increase k, while the descriptive power of our rebuilt dictionary is upper bounded by that of the associated HKM tree.

Our approach naturally provides a confidence measure, so we can quantify the uncertainty in classification as shown in Fig. 4.

Fig. 4. Selected results on Graz-02. (Best viewed in color). Images are first masked by the classification then transformed to HSV. The HSV images have their V channel weighted by the confidence in the classification, darkening the pixels which are less confident about the class. All images shown were generated with the parameter set denoted AIB200RGB and classified with an SVM.

5 Conclusions and Future Work

We have described and shown that an object localization framework which uses bag-of-features as a tool can successfully localize objects without making assumptions about the shape of the object, or explicitly performing segmentation.

In order to make this possible, we have also shown a method that efficiently learns a dictionary which is tailored for the task of categorization. In spite of its simplicity, our approach produces pixel-accurate object localizations which exceed the state of the art on a challenging dataset.

Our experiments show that more care should be exercised in integrating shape information into generic object class representations. We believe shape is an important discriminant ([19], Theorem 3), but our work should be viewed as a baseline method whose performance should be convincingly exceeded before justifying the additional complexity a shape-based model might bring.

The techniques we describe can be directly extended to pixel-level multi-class localization, and we plan to do this. We will also explore adding a notion of scale, perhaps by simply performing multiple classifications at different scales followed by scale selection. We note that in our framework this does not add any significant computational burden since our complexity is not tied to the size of the windows we choose. Our system is already very fast, and we plan to improve the speed further until the system operates in real-time.

Last, our approach could be combined with conditional random fields or other models that are capable of enforcing spatial consistency and context-type constraints (e.g. [20,16]). However, we note that we already have some local consistency built-in since each windowed histogram we classify has a very high overlap with its neighbors.

Acknowledgements

This research was supported by ONR N00014-08-1-0414, 67F-1080868 and AFOSR FA9550-06-1-0138.

References

1. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: Proc. CVPR (2006)
2. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proc. CVPR (2007)
3. Tuytelaars, T., Schmid, C.: Vector quantizing feature space with a regular lattice. In: Proc. ICCV (2007)
4. Marszałek, M., Schmid, C.: Accurate object localization with shape masks. In: Proc. CVPR (2007)
5. Opelt, A., Pinz, A.: Object localization with boosting and weak supervision for generic object recognition. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds.) SCIA 2005. LNCS, vol. 3540, pp. 862–871. Springer, Heidelberg (2005)
6. Viola, P., Jones, M.: Robust real-time object detection. In: Second International Workshop on Statistical and Computational Theories of Vision, Vancouver, Canada (2001)
7. Slonim, N., Tishby, N.: Agglomerative information bottleneck. In: Proc. NIPS (1999)

8. Lazebnik, S., Raginsky, M.: Learning nearest-neighbor quantizers from labeled data by information loss minimization. In: Proc. Conf. on Artificial Intellligence and Statistics (2007)
9. Leibe, B., Micolajckzyk, K., Schiele, B.: Efficient clustering and matching for object class recognition. In: Proc. BMVC (2006)
10. Winn, J., Criminisi, A., Minka, T.: Object categorization by learned universal visual dictionary. In: Proc. ICCV (2005)
11. Marszałek, M., Schmid, C.: Spatial weighting for bag-of-features. In: Proc. CVPR (2006)
12. Leordeanu, M., Hebert, M., Sukthankar, R.: Beyond local appearance: Category recognition from pairwise interactions of simple features. In: Proc. CVPR (2007)
13. Ling, H., Soatto, S.: Proximity distribution kernels for geometric context in category recognition. In: Proc. CVPR (2007)
14. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bag of features: Spatial pyramid matching for recognizing natural scene categories. In: Proc. CVPR (2006)
15. Cao, L., Fei-Fei, L.: Spatially coherent latent topic model for concurrent object segmentation and classification. In: Proc. ICCV (2007)
16. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in context. In: Proc. ICCV (2007)
17. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with implicit shape model. In: ECCV Workshop on Statistical Learning in Comp. Vision (2004)
18. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model (2007), `http://people.cs.uchicago.edu/pff/papers/`
19. Vedaldi, A., Soatto, S.: Features for recognition: Viewpoint invariance for non-planar scenes. In: Proc. ICCV (2005)
20. Shotton, J., Winn, J., Rother, C., Criminisi, A.: TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 1–15. Springer, Heidelberg (2006)
21. He, X., Zemel, R., nán, M.C.P.: Multiscale conditional random fields for image labeling. In: Proc. CVPR (2004)
22. Liu, J., Shah, M.: Scene modeling using co-clustering. In: Proc. ICCV (2007)
23. Agarwal, A., Triggs, B.: Hyperfeatures - multilevel local coding for visual recognition. Technical report, INRIA (2005)
24. Lampert, C., Blaschko, M., Hofmann, T.: Beyond sliding windows: Object localization by efficient subwindow search. cvpr (2008)
25. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: CVPR (2008)
26. Wang, X., Doretto, G., Sebastian, T., Rittscher, J., Tu, P.: Shape and appearance context modeling. In: Proc. ICCV (2007)
27. Porikli, F.: Integral histogram: A fast way to extract histograms in cartesian spaces. In: Proc. CVPR (2005)
28. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. IJCV (2006)
29. Moosmann, F., Triggs, B., Jurie, F.: Fast discriminative visual codebooks using randomized clustering forests. In: Proc. NIPS (2006)
30. Slonim, N.: Iba_1.0: Matlab code for information bottleneck clustering algorithms (2003), `http://www.princeton.edu/nslonim/`

31. Vedaldi, A., Fulkerson, B.: Vlfeat: Feature extraction library (2007), http://vision.ucla.edu/vlfeat/
32. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 2(60), 91–110 (2004)
33. Bay, H., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
34. Tola, E., Lepetit, V., Fua, P.: A fast local descriptor for dense matching. In: Proc. CVPR (2008)
35. Elgammal, A., Harwood, D., Davis, L.: Non-parametric model for background subtraction. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 751–767. Springer, Heidelberg (2000)

Weakly Supervised Object Localization with Stable Segmentations

Carolina Galleguillos[1], Boris Babenko[1],
Andrew Rabinovich[1], and Serge Belongie[1,2]

[1] Computer Science and Engineering, University of California, San Diego
[2] Electrical Engineering, California Institute of Technology
{cgallegu,bbabenko,amrabino,sjb}@cs.ucsd.edu

Abstract. Multiple Instance Learning (MIL) provides a framework for training a discriminative classifier from data with ambiguous labels. This framework is well suited for the task of learning object classifiers from weakly labeled image data, where only the presence of an object in an image is known, but not its location. Some recent work has explored the application of MIL algorithms to the tasks of image categorization and natural scene classification. In this paper we extend these ideas in a framework that uses MIL to recognize and *localize* objects in images. To achieve this we employ state of the art image descriptors and multiple stable segmentations. These components, combined with a powerful MIL algorithm, form our object recognition system called MILSS. We show highly competitive object categorization results on the Caltech dataset. To evaluate the performance of our algorithm further, we introduce the challenging Landmarks-18 dataset, a collection of photographs of famous landmarks from around the world. The results on this new dataset show the great potential of our proposed algorithm.

1 Introduction

The goal of object categorization is to locate and identify instances of an object category within an image. This task is challenging in real world scenes since objects may vary in scale, position, and viewpoint; in addition, they may be surrounded by background clutter, occluded by other objects, and obscured by poor image quality. To model these sources of variability, traditional approaches to object categorization require large labeled data sets of fully annotated training images. Typical annotations in these "fully" labeled data sets provide masks or bounding boxes that specify the locations, scales, and orientations of objects in each training image. Though extremely valuable, this information is prone to error and is expensive to obtain. Without this information, however, traditional approaches to object categorization tend to learn spurious models of background artifacts, leading to lower accuracy during testing.

Some approaches for object categorization have successfully learned object models from weakly labeled data [1,2,3,4,5]. Weakly labeled training examples indicate which objects of interest are present in training images without specifying the pixels that are associated with them. From weakly labeled examples,

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 193–207, 2008.

the existing methods use standard techniques in statistical learning to model the essence of each category. Popular approaches include part-based models [1,6,7], region based methods [2,5] and latent models such as pLSA and LDA, with bag of visual words [3,4,8]. While they excel at exploiting correlations between different image patches, they suffer from computationally expensive inference and background noise that is learned as part of the category model.

Recently, Multiple Instance Learning (MIL) models have been applied to image categorization [9,10]. MIL permits weakly labeled images for training, but avoids the shortcomings of the methods mentioned above. In particular, MIL trains a discriminative classifier, rather than a generative model, which avoids complex inference procedures, and usually results in higher recognition accuracy. Although some of the previous works have applied MIL algorithms to the problem of object categorization, the focus has been on classifying images rather than localizing instances of objects in them.

Following this promising line of work we extend the current frameworks for MIL-based image categorization by adding object localization capabilities and improving image categorization accuracy. The main contribution of this paper is a novel object categorization framework that localizes objects in cluttered, real world scenes. Our method incorporates multiple stable segmentations and Bag-of-Features (BoF) image representation into a MIL framework, see Fig. 1 for an illustration. We demonstrate the efficiency and accuracy of our framework on two databases that present significant intra-class variation: Caltech 4 [11] and a landmark image database, Landmarks-18. The Caltech dataset, although highly popular in the computer vision community, is a rather artificial dataset, where objects often appear in isolation and with uniform backgrounds. The Landmarks-18 dataset on the other hand, is taken directly from common web albums and contains instances of popular landmarks in cluttered scenes with variable viewpoint, weather, and illumination (see Fig. 3).

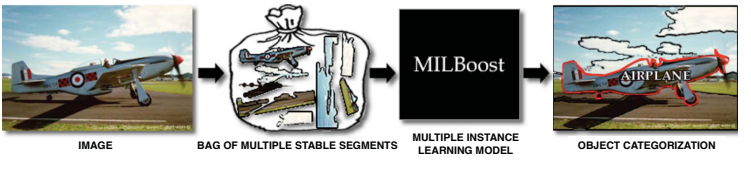


Fig. 1. An input image containing an airplane is processed through a segmentation-based object recognition engine obtaining a collection of stable segments. The bag of segments is represented as bags of features and then fed into the MIL algorithm. Finally, the model classifies each segment, localizing the object in the image.

2 Related Work

2.1 Multiple Instance Learning

The MIL problem was first introduced by Dietterich *et al.* [12] for the problem of drug discovery. In this domain it is desired to predict properties of a drug

molecule using the molecule's shape as an input to the classifier. Each molecule, however, can take on multiple shapes, and it is not known during training which shape is responsible for certain properties of the training molecules. Formally, traditional supervised learning requires training data $\{(x_1, y_1), ..., (x_N, y_N)\}$, $x_i \in \mathcal{X}, y_i \in \mathcal{Y}$ where $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the output space. On the other hand, MIL is able to learn from training data of the form $\{(X_1, y_1), ..., (X_N, y_N)\}, X_i = \{x_{i1}, x_{i2}...\}, x_{ij} \in \mathcal{X}, y_i \in \mathcal{Y}$. For example, in the drug discovery problem each X_i is a molecule, and each x_{ij} is one particular shape of that molecule. The MIL problem is defined only for binary classification, so we will assume that $\mathcal{Y} = \{1, 0\}$. In this setting X_i is an unordered set of inputs (often called a "bag"), and the bag label y_i follows the rule $y_i = \max_j(y_{ij})$. Notice that although true instance labels y_{ij} are assumed to exist, the learning algorithm does not have access to them during training. The goal of a MIL algorithm is then to learn a classifier function $H : x \to \{0, 1\}$, that acts on instances. Various algorithms have been proposed for solving this problem [12,13,14], and in this paper we chose the MilBoost algorithm by Viola *et al.* [14].

2.2 MIL and Image Categorization

In recent years MIL algorithms have attracted the attention of the computer vision community because they provide a way of training classifiers with weakly labeled data. These models have tried to address various problems such as scene classification, image annotation, and image and object categorization. In natural scene classification, several models have successfully classified images into predefined semantic concepts (categories) using MIL. For example, Maron *et al.* applied the Diverse Density (DD) algorithm to the problem of natural scene classification [15]. Trying to solve the same problem, Zhou [16] introduced MIML, where each training example is associated with not only multiple instances but also multiple class labels. Both methods consider classification on a bag (image) level only, and do not take advantage of the instance classifier returned by a MIL algorithm. Similarly, in image annotation, MI-SVM [17] and ASVM-MIL [18] algorithms use variations of the popular SVM algorithm modified to solve MIL. In the problem of image categorization many MIL approaches have been shown to outperform traditional supervised object categorization models. The DD-SVM [19] model uses the DD algorithm to select prototypes and an SVM to classify bags in the prototypes' space. Bi *et al.* [20] and MILES [9] embed bags into a feature space defined by instances and use a 1-norm SVM to construct bag classifiers. Recently, the results of ConMIL [10] showed that modeling interdependencies between instances can improve accuracy in instance and bag classification.

With respect to object categorization, ConMIL and MILES have achieved competitive results relative to traditional approaches. In these methods, object categorization is framed as binary classification which tries to separate object instances from background clutter. Although these algorithms achieve good performance on an image level, their models often capture parts of the background in the positive images. While the backgrounds of positive images provide clues in

image classification (*e.g.* an airplane will often co-occur with a sky background), models that capture this information would have trouble in correctly localizing the objects of interest.

3 Multiple Instance Learning Using Stable Segmentations

The problem of learning an object classifier from weakly labeled data can be elegantly framed as multiple instance learning. During training it is known for each image whether a certain object category is present, but the exact location of that object is unknown. If we split an image $\mathcal{I}_i$ into J multiple regions or segments $\{s_{i1}, s_{i2}..., s_{iJ}\}$, we can assume that one of the segments contains the object of interest (we will discuss different strategies for doing this shortly). For each image we are given a category label $y_i = \{c_1, c_2, ..., c_C\}$; however, since the MIL problem is defined only for binary classification, we will train our classifiers in a one versus all manner. If we define $y_{ik} \in \mathbf{1}(y_i = c_k)$ to be a binary label indicating the presence of category k in image i, we can train C different classifiers. For each category k, we train a classifier $H^k : s \rightarrow \{1, 0\}$ using the training data set $\{(\mathcal{I}_1, y_{ik}), ...\}$. In practice, since our problem is multi-class it is more useful for us to also obtain the probability of the segment containing an object category k, $p(c_k|s)$. The boosting algorithm for MIL developed in [14] provides us with an effective way of learning these functions, and in the section below we briefly review this algorithm.

3.1 MilBoost

The MilBoost algorithm developed by Viola et al. in [14] uses the gradient boosting framework of Friedman [21]. The classifier learned by a boosting framework has the form $H^k(s) = \sum_{t=1}^{T} \alpha_t^k h_t^k(s)$ where each h_t^k is a weak classifier and α_t^k is a scalar weight. We use a simple decision stump as the weak classifier as is done in much of the boosting literature [22,23][1]. To get a binary label from this classifier we could use sign(H^k), but recall that we would also like to retrieve a probability. Instead, we use the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ to define this probability as follows:

$$p(c_k|s) = \sigma\Big(\sum_{t=1}^{T} \alpha_t^k h_t^k(s)\Big). \tag{1}$$

The loss function we optimize is the binomial log likelihood over bags:

$$\mathcal{L}^k(H^k) = -\sum_i \Big(y_{ik} \log(p_{ik}) + (1 - y_{ik}) \log(1 - p_{ik})\Big), \tag{2}$$

where $p_{ik} = p(c_k|\mathcal{I}_i)$ is the probability that image i contains an object from category k. Note that it is impossible to compute the likelihood over segments

[1] Using a decision stump as a weak classifier also results in feature selection during training.

because the labels for these are unknown during training. Finally, we need to define the image probability p_{ik} in terms of the probabilities of its segments. Ideally we would define this as $p_{ik} = \max_j p(c_k|s_{ij})$. Since the boosting framework uses gradient descent to learn, however, this definition would cause problems due to the non differentiable max operator. Instead Viola et al. suggest using the Noisy-OR model as follows:

$$p(c_k|\mathcal{I}_i) = 1 - \prod_j \big(1 - p(c_k|s_{ij})\big). \tag{3}$$

Having all of these terms defined, we can now use the gradient boosting framework to learn each weak classifier h_t^k in a greedy fashion. Given an incomplete classifier $H_{t-1}^k(s) = \sum_{l=1}^{t-1} \alpha_l^k h_l^k(s)$ we seek to add one more weak classifier and its corresponding weight to optimize the overall loss function:

$$(\alpha_t^k, h_t^k) = \underset{(h,\alpha)}{\operatorname{argmin}} \Big(\mathcal{L}^k(H_{t-1}^k + \alpha h)\Big). \tag{4}$$

To achieve this, Viola et al. follow Friedman's suggestion of viewing the boosting procedure as a gradient descent in function space (where the value of H^k for every training instance corresponds to a dimension). In this sense, we would like to add a weak classifier h_t^k that is along the direction of the gradient

$$w_{ij} = \frac{\partial \mathcal{L}^k}{H^k(s_{ij})}\bigg|_{H_{t-1}^k}. \tag{5}$$

Unfortunately, we cannot move in arbitrary directions in function space because we are limited by the class of weak learners we have chosen. Therefore, we would like to choose a weak classifier which moves in a direction that is as close as possible to this gradient:

$$h_t^k = \underset{h}{\operatorname{argmin}} \sum_{ij} h(s_{ij}) w_{ij}. \tag{6}$$

Finally, we can determine α_t by doing a simple line search.

3.2 Region Extraction

In MIL an image is divided into segments or regions and each region is represented by a high dimensional feature vector. Existing MIL-based approaches have adopted a variety of techniques for partitioning an image, including blocks, patches and single segmentations. One simple convention is to use a single non overlapping grid of 4×4 blocks [9,19,20]. In order to obtain representative regions of the possible objects in the scene, this block segmentation is followed by K-means clustering of the feature vectors extracted from the blocks. The number of clusters depends on the the number of objects that typically appear in the scene, introducing a model order selection problem. Other MIL-based

approaches [9,10] extract salient regions using Kadir's detector [24]. This allows them to compare their object categorization results to non-MIL-based methods. Salient regions are detected over different locations and scales and then cropped from the image and rescaled into an image patch of size 11×11 pixels. Partitioning an image into blocks or patches often breaks an object into several pieces or puts different objects into a single patch. Alternatively, image segmentation is a way to decompose an image into a collection of regions that hopefully correspond to objects. The methods in [15,17] use the blobworld representation of [25] in which an image is segmented into a set of regions, each characterized by color, texture and shape descriptors. Other aproaches [10,18] obtain meaningful image regions using JSEG [26] or NCut [27] segmentation algorithms. Each image is typically segmented into ten or fewer regions, and only segments bigger than a certain threshold are kept. However, as described in [28], there usually does not exist a single correct segmentation of an image, but rather a collection of potentially meaningful image segmentations. Thus, using just a single segmentation may hinder recognition due to splitting or merging errors.

The idea of using multiple segmentation has recently emerged [3,5,29,30,31] in the area of object recognition. Segmentations are computed resulting in a bag (or soup) of segments, with the hope that a subset of them will capture adequate object boundaries. Multiple stable segmentations have been shown to produce competitive results in object categorization [29,32]. In this work we advocate their use as a substrate for MIL-based object categorization.

3.3 Multiple Stable Segmentations

In order to extract more adequate image regions for our system, we compute multiple stable segmentations [28]. The method of multiple stable segmentations uses stability as a heuristic for a particular set of parameters, cue weightings and a model order. For each choice of parameters for cue combinations $\boldsymbol{p}$ and number of segments q, the image is segmented using Normalized Cuts [27,33]. The segmentation is considered stable if small perturbations of the image do not yield substantial changes in the segmentation. The image is perturbed and segmented T times and the following score is evaluated:

$$\Phi(q, \boldsymbol{p}) = \frac{1}{n - \frac{n}{q}} \left(\sum_{i=1}^{n} \sum_{j=1}^{T} \delta_{ij} - \frac{n}{q} \right), \text{where } \delta_{ij} = \begin{cases} 1 & \text{if } i = \text{j} \\ 0 & \text{otherwise} \end{cases}. \quad (7)$$

Here n is the number of pixels and δ_{ij} is equal to 1 if the i-th pixel is mapped to a different segment in the j-th perturbed segmentation, and zero otherwise. Thus Φ is a properly normalized[2] measure of the probability of a pixel to change label due to a perturbation of the image. Segmentations with a high stability score are retained. Notice that, in general, there may exist several stable segmentations for an image.

[2] In particular Φ ranges in $[0, 1]$ and it is not biased towards a particular value of q.

3.4 MILSS Framework

Our Multiple Instance Learning framework using multiple Stable Segmentations (MILSS) presents a novel approach for object categorization that combines popular elements from previous work in object recognition with a MIL framework. Multiple stable segmentations [28] provide a spatial grouping of pixels into regions that increase the chances of extracting meaningful segments for MIL. They are memory efficient compared to extracting a large number of patches and they provide localization capability to our framework.

In order to improve instance classification, we use the bag of features model (BoF) [11] to capture appearance information. Recently, the BoF image representation has found widespread application in object categorization due to its simplicity and efficiency. To represent an image segment as a BoF, we first detect salient regions in the segment and compute a feature vector for each region. These feature vectors are then mapped to a vocabulary of "visual words" which are computed using vector quantization. The BoF representation of an image segment is then a histogram of these visual words (often referred to as a signature).

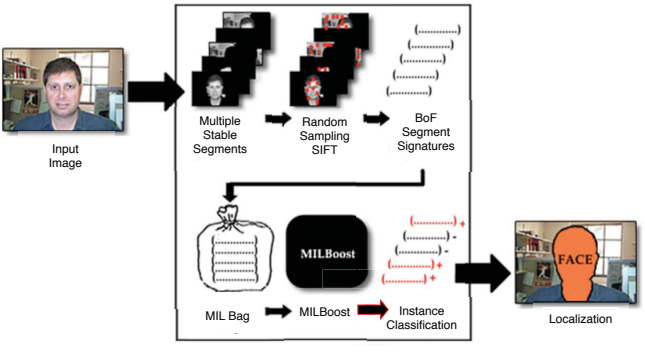


Fig. 2. An object is recognized by the MILSS framework. An input image containing a face is partitioned into a collection of stable segments. Then a BoF approach computes SIFT [34] descriptors in a random fashion on each segment. A signature is computed for each segment and the resulting bag of signatures is fed into the MILBOOST model. MILBOOST classifies each signature (instance) and the bag, resulting in the localization of the face within the image and the classification of the image as a whole.

We combine multiple segmentations and the BoF representation with the MILBOOST framework [14] which performs feature selection during training and allows rapid segment and image classification at runtime. Figure 2 shows each step of our categorization model. Next we address, in detail, how the image segments and their signatures are used for object categorization.

Classification. Given an image $\mathcal{I}_i$ we compute q stable segmentations resulting in multiple segments $\{s_{i1}, s_{i2}..., s_{iJ}\}$. For each segment s_{ij} we compute a BoF signature, with each signature corresponds to an instance of the bag. A segment s_{ij} is classified as follows:

$$y_{ij} = \underset{k}{\operatorname{argmax}}\, p(c_k|s_{ij}), \tag{8}$$

where $p(c_k|s_{ij})$ is the probability of the segment s_{ij} belonging to the category c_k, defined by Eq. 1. We classify an image $\mathcal{I}_i$ as proposed by [29]:

$$y_i = \underset{k}{\operatorname{argmax}} \sum_{j=1}^{J} p(c_k|s_{ij}). \tag{9}$$

Localization. The task of object localization generally corresponds to placing a bounding box, or preferably the actual object outline, around the object within the image. Since our framework uses segments for categorization, we utilize segment boundaries that yield highest recognition score in order to describe object locations [1]. For evaluating our localization performance for an image $\mathcal{I}_i$ classified overall as y_i and segment labels y_{ij}, we look for segments with labels such that $y_{ij} = y_i$. Then we check for overlapping segments and return the first n unique segment boundaries, with $n \ll J$.

4 Experimental Results

To evaluate the MILSS framework, we compare our approach to the state-of-the-art methods in object categorization. Existing MIL-based approaches often use the COREL dataset to evaluate their models for image categorization. However, since we concentrate on object categorization, the performance of our approach is evaluated on Caltech 4 and a new dataset Landmarks-18.

4.1 Caltech 4 Dataset

Caltech 4 [11] is a well established dataset and is a standard benchmark for object categorization. Although simple, we utilize this dataset as a means of comparison with Mil-based methods. Following the experimental set up of [9,10], we perform a category versus background classification. Table 1(a) presents the results of categorization accuracy for our method. Results are compared to existing MIL-based image categorization models [9,10] and a non-MIL-based approach of [6]. The presented results are competitive with the rest of the algorithms. The average categorization accuracy for MILSS as well as ConMIL is 98%; while MILES is 97% and Bar-Hillel *et al.*'s algorithm is 93%. Note that the highest performance is achieved in the Airplanes category given that the stable segmentations were able to separate the background from the objects accurately. In a second experiment, we include the Leopard class for comparing our method to existing algorithms [8,11] in a multi-class setting.

Table 1(b) reports accuracy for multi-class object categorization. Instead of considering a background category, images belonging to each category acted as negative examples for models trained on the other categories. We compare our method to existing non-MIL-based object recognition frameworks: the dependent Hierarchical Dirichlet process (DHDP) [8] and constellation of parts model

Table 1. (a) Comparison of categorization results between our framework, MIL-based models [9,10] and a traditional object categorization approach [6] for Caltech 4 categories. Results in **bold** indicate the highest performance for each category. (b) MILSS Confusion matrix between the four categories for multi-class object recognition.

(a)

	Airplanes	Cars	Faces	Motorbikes
Training data	400	400	218	400
MILSS	**1**	.971	.976	.972
ConMIL [10]	.992	**.984**	.976	**.987**
MILES [9]	.980	.945	**.995**	.967
Bar-Hillel [6]	.897	.977	.917	.931

(b)

	A	F	L	M
Training data	400	218	100	400
Airplanes (A)	**.98**	.00	.01	.01
Faces (F)	.01	**.99**	.00	.00
Leopards (L)	.05	.01	**.93**	.01
Motorbikes (M)	.01	.00	.01	**.97**

[1]. As shown in Table 2(a), MILSS reports an average recognition accuracy of 97% while DHDP reports 98%. Looking closely at the categories, MILSS outperforms DHDP in three out of four of them. The Leopards category seems to be the most challenging for our framework, since it contains fewer images than the rest of the categories (100 for training and 100 for testing). In order to improve these results we could easily augment our training set with images from public repositories, as manual labeling is not required.

Table 2. (a) Results of multiple object categorization models for four Caltech categories. We compare our results to those of non MIL-based models. Results in **bold** indicate the highest performance for each category. (b) Average localization results of MILSS for four categories of Caltech.

(a)

	Airplanes	Faces	Leopards	Motorbikes	Mean
Training data	400	218	100	400	
MILSS	**.977**	**.986**	.927	.971	0.965
DHDP [8]	.961	.978	**1**	.967	0.976
Fergus [1]	.888	.862	-	**.977**	0.909

(b)

	MILSS
Airplanes	.932
Faces	.902
Leopards	.891
Motorbikes	.859
Mean	.896

In a multi-class setting, localization accuracy of MILSS is 90% on the Caltech 4 dataset. Our localization results are presented in detail in Table 2 (b). To quantify the accuracy of object localization we adopt the methodology of [1] and consider the overlap $\alpha = \frac{B \cap B_{gt}}{B \cup B_{gt}}$. Note that our method may be at a disadvantage in cases where the objects' contour areas B are smaller than the ground truth bounding box B_{gt}; thus it is difficult to make a direct comparison with the results in [1]. Since our method localizes objects using segment boundaries, the location and extent of the object is captured more precisely than those with bounding boxes, see Fig. 6.

4.2 Landmark Database

With the increasing popularity of digital photography and the user's desire to share their pictures in web albums, recognition of destinations and landmarks

Fig. 3. Landmarks-18 Dataset. Two examples are shown per landmark and each row shows 9 categories. **Top row**: Arc de Triomphe, Ayres Rock, Bellsouth Building, Brandenburg Gate, Buckingham Palace, Burjal Arab, CN Tower, Centre Pompidou and Chrysler Building. **Bottom row**: Church Savior Spilled Blood, Eiffel Tower, Liberty Bell, Lincoln Memorial, Lincoln Memorial Statue, London Tower Bridge, Space Needle, Sydney Opera House and Taipei 101.

has become an interesting problem. Recognizing objects in real world images is a challenging task, as images are presented at a variety of viewpoints, scales, and illuminations; noise, background clutter, and occlusions also make the problem more difficult. Since photo-sharing sites are a vast resource of weakly labeled image data, we easily gather large datasets to evaluate our framework.

In this paper we introduce a new dataset called Landmarks-18, consisting of 18 different categories of landmarks, provided by Google Research and collected from public web albums. Landmarks-18 captures much more significant intra-class variability than standard benchmark datasets for object recognition. Figure 3 demonstrates the diversity of landmarks in the dataset while Fig. 5(b) provides the statistics of the dataset.

Here we performed two different multi-class categorization experiments on Landmarks-18. Each experiment considers 10 different categories, where images in each category were divided randomly into 80%/20% for training and testing respectively. Experiments were performed with 5-fold cross validation to obtain statistically relevant average categorization results. Figure 4 shows confusion matrices for both experiments. The results show that Landmarks-18 is much more difficult for categorization than Caltech 4, due to the challenging characteristics of its images and the larger number of classes. Despite this, MILSS achieves high categorization accuracy in both experiments. The outcome of both experiments indicate that Eiffel Tower, Taipei101, and Bellsouth Building are the most challenging categories. The main source of low recognition accuracy isnbetween visually similar categories such as Bellsouth Building vs. Chrysler

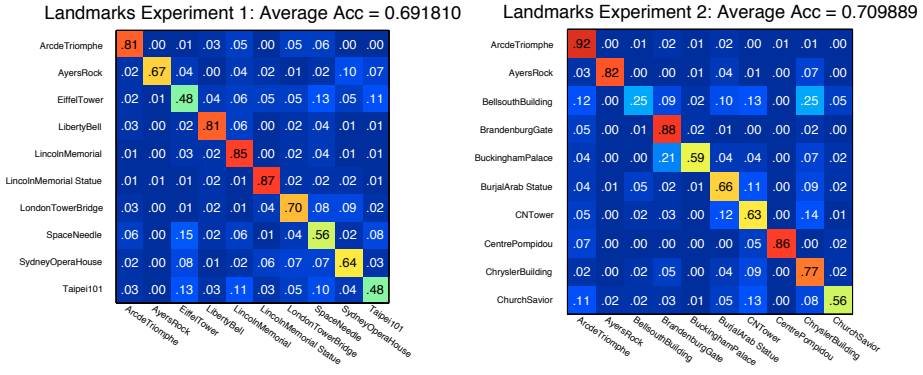
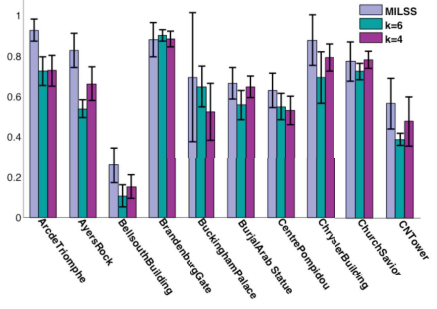


Fig. 4. Confusion matrices of categorization accuracy for the Landmark-18 dataset. (a) Experiment 1; (b) Experiment 2.

Building. For this dataset we were unable to compare our results to other MIL-based categorization systems as code was not available.

To evaluate the importance of the multiple stable segmentations within MILSS, we also experimented with two different single segmentations ($q = 4$ and $q = 6$) using Normalized Cuts [27]. Figure 5 (a) shows the average categorization accuracy for each method using 5-fold cross validation. With multiple stable segmentations categorization performance is improved in almost all categories. The average categorization accuracy for $q = 4, 6$ and multiple segmentations is 58.3%, 61.8% and 71.0% respectively. The total number of segmentations extracted from an image plays an important role in categorization accuracy. As noted by others, as the number of segments per image increases, so does the



(a)

Category	n	Category	n
ArcdeTriomphe	146	ChurchSavior	109
AyresRock	113	EiffelTower	194
BellsouthBuild	107	LibertyBell	175
BrandenburgG	166	LincolnMem	198
BuckinghamP	87	LincolnMStatue	200
BurjalArab	158	LondonTower	195
CNTower	160	SydneyOHouse	186
CPompidou	71	SpaceNeedle	219
ChryslerBuild	204	Taipei101	176

(b)

Fig. 5. (a) Three different types of region extraction: two single segmentations with number of segments equal to 4 and 6, and multiple stable segmentations. The average categorization accuracy for $q = 4, 6$ and multiple segmentations is 58.3%, 61.8% and 71.0% respectively. Multiple stable segmentations outperform (on average) all the other methods. (b) shows the statistics of Landmarks-18 database.

chance of having a segment that represents the object accurately [29,30]. We believe that multiple stable segmentations provide a way of gathering the most meaningful segments, as is reflected in our results.

4.3 Implementation Details

The stability based image segmentation was implemented using Normalized Cuts [27,35]. Five iterations, combining brightness and texture cues with $\boldsymbol{p} = \{0.4, 0.5, 0.6, 0.7\}$ were used to sample the parameter space. For the categorization experiments done for Caltech and Landmarks-18, we computed 5 different

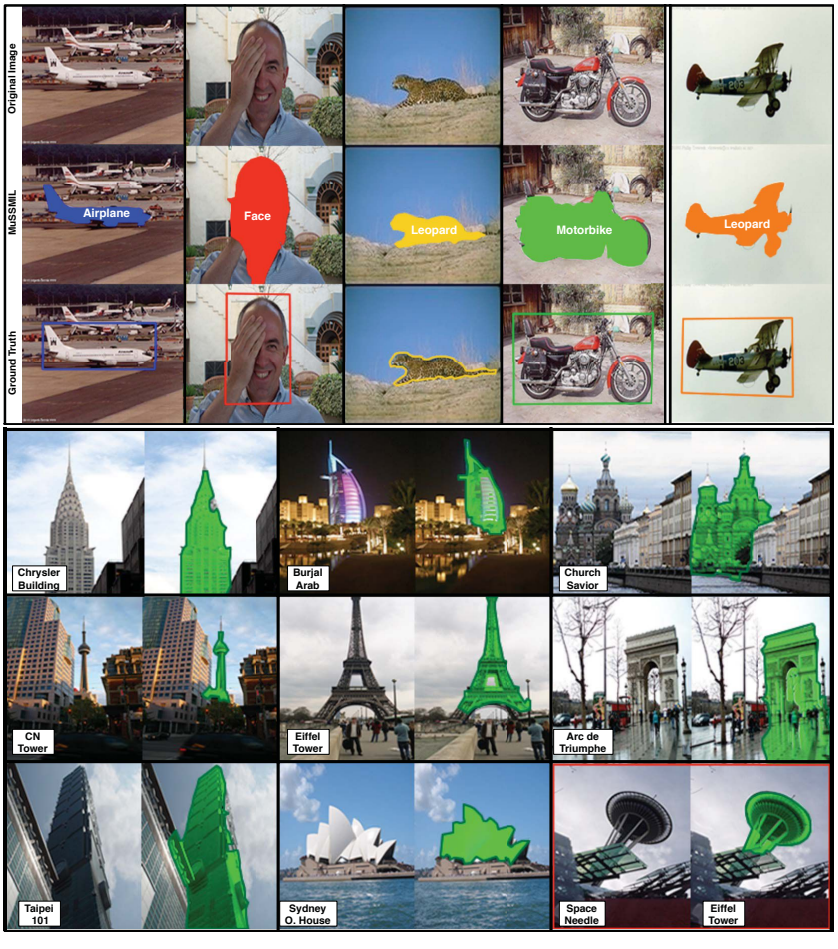


Fig. 6. Top image: Examples of Caltech test images. First three columns correspond to successful image categorization and localization of objects in the scene. Last column correspond to a false positive. **Bottom row**: Examples of Landmarks-18 test images. Green segments represent the image region with the highest probability of being the landmark. Images enclosed by a red rectangle correspond to a false positive.

segmentations with $q = 2, \ldots, 6$ with a total of 20 segments per image. Computing a single segmentation takes about 20-30 seconds per image. For the BoF model we computed 5000 random SIFT [34] features at multiple scales (from 12 pixels up to the full image size) for each image segment. Visual words are obtained computing a hierarchical K-means with $K = 17$ and three levels. The computation of SIFT descriptors and signatures takes about 1 second per segment in a MATLAB/C implementation. Constructing the vocabulary tree takes 40-50 minutes for ten categories. Training time for MilBoost on four Caltech categories takes about 1 day using 500 weak classifiers. Using ten categories of Landmarks-18 MilBoost take less than a day of training using 200 weak classifiers. Classification of all test images for ten categories is done in 0.5 seconds. All above operations were performed on a Pentium 2.8 GHz.

5 Conclusions and Future Work

In this paper we proposed a novel framework for image categorization and localization of objects in real world scenes using weakly labeled data. Our performance is highly competitive with current MIL-based and traditional approaches for image and object categorization. We showed that multiple stable segmentations extracted suitable regions for the MIL problem, thus increasing performance in categorization and permitting accurate localization capabilities. We tested our framework on Caltech 4 and Landmarks-18 datasets, obtaining high accuracy in object categorization tasks. As future work, we want to explore new methods to scale our object categorization framework to a larger number of categories and handle multiple objects in the scene.

Acknowledgments. Special thanks to Hartmut Neven and Hartwig Adam from Google for providing the Landmarks-18 dataset used in this paper. This work was funded in part by NSF Career Grant #0448615, the Alfred P. Sloan Research Fellowship, NSF IGERT Grant DGE-0333451 and a Google Research Award.

References

1. Fergus, R., Perona, P., Zisserman, A.: Weakly supervised scale-invariant learning of models for visual recognition. IJCV 71(3), 273–303 (2007)
2. Opelt, A., Fussenegger, M., Auer, P.: Generic object recognition with boosting. PAMI 28(3), 416–431 (2006)
3. Russell, B., Efros, A., Sivic, J., Freeman, W., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: CVPR (2006)
4. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering object categories in image collections. In: CVPR (2005)
5. Todorovic, S., Ahuja, N.: Extracting subimages of an unknown category from a set of images. In: CVPR (2006)
6. Bar-Hillel, A., Hertz, T., Weinshall, D.: Object class recognition by boosting a part-based model. In: CVPR (2005)

7. Crandall, D., Huttenlocher, D.: Weakly supervised learning of part-based spatial models for visual object recognition. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 16–29. Springer, Heidelberg (2006)
8. Wang, G., Zhang, Y., Fei-Fei, L.: Using dependent regions for object categorization in a generative framework. In: CVPR (2006)
9. Chen, Y., Bi, J., Wang, J.: MILES: Multiple-instance learning via embedded instance selection. PAMI 28(12), 1931–1947 (2006)
10. Qi, G., Hua, X., Rui, Y., Mei, T., Tang, J., Zhang, H.: Concurrent multiple instance learning for image categorization. In: CVPR (2007)
11. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR (2003)
12. Dietterich, T.G., Lathrop, R.H., Perez, L.T.: Solving the multiple-instance problem with axis parallel rectangles. AAAI, Menlo Park (1997)
13. Andrews, S., Hofmann, T., Tsochantaridis, I.: Multiple instance learning with generalized support vector machines. AAAI, Menlo Park (2002)
14. Viola, P., Platt, J.C., Zhang, C.: Multiple instance boosting for object detection. In: NIPS, vol. 18 (2006)
15. Maron, O., Ratan, A.: Multiple-instance learning for natural scene classification. In: ICML (1998)
16. Zhou, Z., Zhang, M.: Multi-instance multi-label learning with application to scene classification. In: NIPS, vol. 19 (2007)
17. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: NIPS, vol. 15 (2002)
18. Yang, C., Dong, M., Hua, J.: Region-based image annotation using asymmetrical support vector machine-based multi-instance learning. In: CVPR (2006)
19. Chen, Y., Wang, J.: Image categorization by learning and reasoning with regions. JMLR 5, 913–939 (2004)
20. Bi, J., Chen, Y., Wang, J.: A sparse support vector machine approach to region-based image categorization. In: CVPR (2005)
21. Friedman, J.H.: Greedy function approximation: A gradient boosting machine. The Annals of Statistics 29(5), 1189–1232 (2001)
22. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. JCSS 55, 119–139 (1997)
23. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR (2001)
24. Kadir, T., Brady, M.: Saliency, scale and image description. IJCV 45 (2001)
25. Carson, C., Belongie, S., Greenspan, H., Malik, J.: Blobworld: image segmentation using expectation-maximization and its application to image querying. PAMI 24(8), 1026–1038 (2002)
26. Deng, Y., Manjunath, B.: Unsupervised segmentation of color-texture regions in images and video. PAMI 23(8), 800–810 (2001)
27. Shi, J., Malik, J.: Normalized cuts and image segmentation. PAMI 22(8), 888–905 (2000)
28. Rabinovich, A., Lange, T., Buhmann, J., Belongie, S.: Model order selection and cue combination for image segmentation. In: CVPR (2006)
29. Rabinovich, A., Vedaldi, A., Belongie, S.: Does image segmentation improve object categorization? UCSD Technical Report CSE CS2007-0908 (2007)
30. Malisiewicz, T., Efros, A.: Improving spatial support for objects via multiple segmentations. BMVC (2007)

31. Roth, V., Ommer, B.: Exploiting low-level image segmentation for object recognition. In: Franke, K., Müller, K.-R., Nickolay, B., Schäfer, R. (eds.) DAGM 2006. LNCS, vol. 4174, pp. 11–20. Springer, Heidelberg (2006)
32. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewora, E., Belongie, S.: Objects in context. In: ICCV (2007)
33. Malik, J., Belongie, S., Shi, J., Leung, T.: Textons, contours and regions: Cue integration in image segmentation. In: ICCV (1999)
34. Lowe, D.: Object recognition from local scale-invariant features. In: ICCV (1999)
35. Cour, T., Benezit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. In: CVPR (2005)

A Perceptual Comparison of Distance Measures for Color Constancy Algorithms

Arjan Gijsenij, Theo Gevers, and Marcel P. Lucassen

Intelligent Systems Laboratory Amsterdam
University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

Abstract. Color constancy is the ability to measure image features independent of the color of the scene illuminant and is an important topic in color and computer vision. As many color constancy algorithms exist, different distance measures are used to compute their accuracy. In general, these distances measures are based on mathematical principles such as the angular error and Euclidean distance. However, it is unknown to what extent these distance measures correlate to human vision.

Therefore, in this paper, a taxonomy of different distance measures for color constancy algorithms is presented. The main goal is to analyze the correlation between the observed quality of the output images and the different distance measures for illuminant estimates. The output images are the resulting color corrected images using the illuminant estimates of the color constancy algorithms, and the quality of these images is determined by human observers. Distance measures are analyzed how they mimic differences in color naturalness of images as obtained by humans.

Based on the theoretical and experimental results on spectral and real-world data sets, it can be concluded that the perceptual Euclidean distance (PED) with weight-coefficients ($w_R = 0.26$, $w_G = 0.70$, $w_B = 0.04$) finds its roots in human vision and correlates significantly higher than all other distance measures including the angular error and Euclidean distance.

1 Introduction

Color constancy is the ability of a visual system, either human or machine, to maintain stable object color appearances despite considerable changes in the color of the illuminant. Color constancy is a central topic in color and computer vision. The usual approach to solve the color constancy problem is by estimating the illuminant from the visual scene, after which reflectance may be recovered.

Many color constancy methods have been proposed, e.g. [1,2,3,4]. For benchmarking, the accuracy of color constancy algorithms is evaluated by computing a distance measure on the same data sets such as [5,6]. In fact, these distance measures compute to what extent an original illuminant *vector* approximates the estimated one. Two commonly used distance measures are the Euclidean

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 208–221, 2008.

distance and the angular error, of which the latter is probably more widely used than the first. In [7], an analysis is presented of the distribution of these measures, with the aim to find the best summarizing statistic over a large set of images. However, as these distance measures themselves are based on mathematical principles and computed in normalized-rgb color space, it is unknown whether these distance measures correlate to human vision. Further, other distance measures could be defined based on the principles of human vision.

Therefore, in this paper, a taxonomy of different distance measures for color constancy algorithms is presented first, ranging from mathematics-based distances, to perceptual and color constancy specific distances. Then, a perceptual comparison of these distance measures for color constancy is provided. To reveal the correlation between the distance measures and humans, color corrected images will be compared with the original images under reference illumination by visual inspection. In this way, distance measures are evaluated by psychophysical experiments involving paired comparisons of the color corrected images.

The paper is organized as follows. In section 2, color constancy and image transformation is discussed. Further, a set of color constancy methods will be introduced. Then, the different distance measures will be presented in section 3. The first type concerns mathematical measures, including the angular error and Euclidean distance. The second type concerns measuring the distance in different color spaces, e.g. device-independent, perceptual or intuitive color spaces. Thirdly, two domain-specific distance measures are analyzed. In section 4, the experimental setup of the psychophysical experiments is discussed, and the results of these experiments are given in section 5.

2 Color Constancy

The image values $\mathbf{f}$ for a Lambertian surface depend on the color of the light source $e(\lambda)$, the surface reflectance $s(\mathbf{x}, \lambda)$ and the camera sensitivity function $\mathbf{c}(\lambda)$, where λ is the wavelength of the light and $\mathbf{x}$ is the spatial coordinate:

$$\mathbf{f}(\mathbf{x}) = \int_{\omega} e(\lambda)\mathbf{c}(\lambda)s(\mathbf{x}, \lambda)d\lambda, \tag{1}$$

where ω is the visible spectrum. Assuming that the scene is illuminated by one light source and that the observed color of the light source $\mathbf{e}$ depends on the color of the light source $e(\lambda)$ as well as the camera sensitivity function $\mathbf{c}(\lambda)$, then color constancy is equivalent to the estimation of $\mathbf{e}$ by:

$$\mathbf{e} = \int_{\omega} e(\lambda)\mathbf{c}(\lambda)d\lambda, \tag{2}$$

given the image values of $\mathbf{f}$, since both $e(\lambda)$ and $\mathbf{c}(\lambda)$ are, in general, unknown. This is an under-constrained problem and therefore it can not be solved without further assumptions.

2.1 Color Constancy Algorithms

For the purpose of the experiments in this paper, the focus is on a number of simple algorithms. Recently, van de Weijer et al. [1] proposed a framework with which systematically many different algorithms can be constructed. Possible algorithms include methods using 0^{th}-order statistics (i.e. pixel values), like the White-Patch [2], the Grey-World [3] and the Shades-of-Grey algorithms [4], and methods using higher-order (e.g. 1^{st}- and 2^{nd}-order) statistics, like the Grey-Edge and 2^{nd}-order Grey-Edge algorithms. The framework is given by:

$$\left(\int \left| \frac{\partial^n \mathbf{f}_\sigma(\mathbf{x})}{\partial \mathbf{x}^n} \right|^p d\mathbf{x} \right)^{\frac{1}{p}} = k\mathbf{e}^{n,p,\sigma}, \tag{3}$$

where n is the order of the derivative, p is the Minkowski-norm and $\mathbf{f}^\sigma(\mathbf{x}) = \mathbf{f} \otimes \mathbf{G}_\sigma$ is the convolution of the image with a Gaussian filter with scale parameter σ. For the purpose of this article, five instantiations are used, representing a wide variety of algorithms, being the White-Patch ($\mathbf{e}^{0,\infty,0}$), the Grey-World ($\mathbf{e}^{0,1,0}$), the General Grey-World ($\mathbf{e}^{0,13,2}$), the 1^{st}-order Grey-Edge ($\mathbf{e}^{1,1,6}$) and the 2^{nd}-order Grey-Edge algorithm ($\mathbf{e}^{2,1,5}$). Of course, many other algorithms can be generated, but for simplicity, the focus is on these five instantiations as they are derived from different orders of image statistics.

2.2 Image Transformation

Once the color of the light source is estimated, this estimate can be used to transform the input image to be taken under a reference (often white) light source. This transformation can be modeled by a diagonal mapping or *von Kries Model* [8]. The diagonal mapping is given as follows:

$$\mathbf{f}^c = \mathcal{D}^{u,c}\mathbf{f}^u \Rightarrow \begin{pmatrix} R^c \\ G^c \\ B^c \end{pmatrix} = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \gamma \end{pmatrix} \begin{pmatrix} R^u \\ G^u \\ B^u \end{pmatrix}, \tag{4}$$

where $\mathbf{f}^u$ is the image taken under an unknown light source, $\mathbf{f}^c$ is the same image transformed, so it appears if it was taken under the reference light, and $\mathcal{D}^{u,c}$ is a diagonal matrix which maps colors that are taken under an unknown light source u to their corresponding colors under the canonical illuminant c. The diagonal mapping is used throughout this paper to create output-images after correction by a color constancy algorithm.

3 Distance Measures

Performance measures evaluate the performance of an illuminant estimation algorithm by comparing the estimated illuminant to a ground truth, which is known a priori. Since color constancy algorithms can only recover the color of the light source up to a multiplicative constant (i.e. the intensity of the light

source is not estimated), distance measures compute the degree of resemblance in normalized-rgb:

$$r = \frac{R}{R+G+B}\ , \quad g = \frac{G}{R+G+B}\ , \quad b = \frac{B}{R+G+B}. \tag{5}$$

In color constancy research, two frequently used performance measures are the Euclidean distance and the angular error, of which the latter is probably more widely used than the first. The Euclidean distance between the estimated light source $\mathbf{e}_e$ and the true, ground truth, light sources $\mathbf{e}_u$ is given by:

$$\mathcal{L}_2(\mathbf{e}_e, \mathbf{e}_u) = \sqrt{(R_e - R_u)^2 + (G_e - G_u)^2 + (B_e - B_u)^2}. \tag{6}$$

The angular error measures the angular between the estimated illuminant $\mathbf{e}_e$ and the ground truth $\mathbf{e}_u$, and is defined as:

$$d_{\text{angle}}(\mathbf{e}_e, \mathbf{e}_u) = \cos^{-1}\left(\frac{\mathbf{e}_e \cdot \mathbf{e}_u}{||\mathbf{e}_e|| \cdot ||\mathbf{e}_u||}\right), \tag{7}$$

where $\mathbf{e}_e \cdot \mathbf{e}_u$ is the dot product of the two illuminants and $||\cdot||$ is the Euclidean norm of a vector.

Although the value of these two distance measures indicates how closely an original illuminant vector is approximated by the estimated one (after intensity normalization), it remains unclear how these values correspond to human vision. Further, other distances can be derived. To this end, in this section, a taxonomy of different distance measures for color constancy algorithms is presented. The different distance measures are defined ranging from mathematics - based distance measures (section 3.1), to perceptual measures (section 3.2) and color constancy specific measures (section 3.3).

3.1 Minkowski Distance

A well-known measure is the Minkowski distance:

$$\mathcal{L}_p(\mathbf{e}_e, \mathbf{e}_u) = (|R_e - R_u|^p + |G_e - G_u|^p + |B_e - B_u|^p)^{\frac{1}{p}}, \tag{8}$$

where p is the corresponding Minkowski-norm. In this paper, three special cases of this distance measure are evaluated. These three measures are the Manhattan distance ($\mathcal{L}_1$), the Euclidean distance ($\mathcal{L}_2$) and the Chebychev distance ($\mathcal{L}_\infty$).

3.2 Perceptual Distances

The goal of color constancy algorithms, in this paper, is to obtain perceptual distance to a reference image. For this purpose, the estimated color of the light source and the ground truth are first transformed to different (human vision) color spaces, after which they are compared. Therefore, in this section, the distance is measured in the perceptually uniform color spaces $L^*a^*b^*$ and $L^*u^*v^*$ [9], as well as in the more intuitive color channels chroma C and hue H.

Most color constancy algorithms are restricted to estimating the chromaticity values of the illuminant. To evaluate the performance of an estimated light source in different color spaces, this (intensity normalized) estimate, as well as the ground truth light source, is applied to a perfect white reflectance. Hence, two (R, G, B)-values are obtained, representing the color of a white reflectance under the estimated and the true light source. These (R, G, B)-values can consequently be converted to different color spaces. Conversion from RGB to XYZ is done using the following linear transformation:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.4125\ 0.3576\ 0.1804 \\ 0.2127\ 0.7152\ 0.0722 \\ 0.0193\ 0.1192\ 0.9502 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{9}$$

Then, the two perceptual color models $L^*a^*b^*$ and $L^*u^*v^*$ are computed using $(X_w, Y_w, Z_w) = (0.9505, 1.0000, 1.0888)$ as reference white [9]. From these perceptual color spaces, different color channels can be computed, like chroma C and hue H. The transformation from $L^*a^*b^*$ to C and H is given by:

$$C_{ab} = \sqrt{(a^*)^2 + (b^*)^2}\ , \quad H_{ab} = \tan^{-1}\left(\frac{b^*}{a^*}\right), \tag{10}$$

and analogously for $L^*u^*v^*$.

Finally, it is known that the human eye is more sensitive to some colors than to others. This important property of the human visual system is used, for instance, in the conversion of RGB-images to luminance-images [10]:

$$Lum = 0.3R + 0.59G + 0.11B. \tag{11}$$

Hence, a change in the green-channel has a stronger effect on the perceived difference between two images than a change in the blue-channel, for instance. This leads us to the weighted Euclidean distance, or perceptual Euclidean distance (PED). The weights for the different color channels are described as sensitivity measures as follows:

$$\mathrm{PED}(\mathbf{e}_e, \mathbf{e}_u) = \sqrt{w_R(R_e - R_u)^2 + w_G(G_e - G_u)^2 + w_B(B_e - B_u)^2}, \tag{12}$$

where $w_R + w_G + w_B = 1$.

3.3 Color Constancy Distances

In this section, two color constancy specific distances are discussed. The first is the color constancy index CCI [11], also called Brunswik ratio [12], and is generally used to measure perceptual color constancy [13,14]. It is defined as the ratio of the amount of adaptation that is obtained by a human observer versus no adaptation at all:

$$\mathrm{CCI} = \frac{b}{a}, \tag{13}$$

where b is defined as the distance from the estimated light source to the true light source and a is defined as the distance from the true light source to a white reference light.

The second is a new measure, called the *gamut intersection*, that makes use of the gamuts of the colors that can occur under a given light source. It measures the fraction of colors that occur under the estimated light source, with respect to the colors that occur under the true, ground truth, light source:

$$d_{\text{gamut}}(\mathbf{e}_e, \mathbf{e}_u) = \frac{\text{vol}(\mathcal{G}_e \cap \mathcal{G}_u)}{\text{vol}(\mathcal{G}_u)}, \tag{14}$$

where $\mathcal{G}_i$ is the gamut of all possible colors under illuminant i and $\text{vol}(\mathcal{G}_i)$ is the volume of this gamut. The gamut $\mathcal{G}_i$ is computed by applying the diagonal mapping, corresponding to light source i, to a canonical gamut.

4 Experimental Setup

In this section, the experimental setup of the psychophysical experiments is discussed. The experiments are performed on two data sets, one containing hyperspectral recordings of natural and rural scenes, and the other containing a range of real-world scenes. The images are shown on a calibrated monitor, and observers are shown images in a round-robin schedule. For every pair of images, the observers have to specify which of the two results is closer to the ideal result. In this way, comparison of the distance measures (objective performance) is compared with visual judgment (subjective performance) by computing the correlation between the two performance measures.

4.1 Data

Two data sets are used for the psychophysical experiments. The first data set consists of hyperspectral images and is used to perform a thorough, i.e. colorimetrically correct, analysis. The second data set consists of real-world images and is used to analyze the results of the first experiments.

Hyperspectral data. The first data set, originating from [14] consists of eight hyperspectral images, of which four are shown in figure 1(a)-(d). These images were chosen in order to be able to study realistic, i.e. colorimetrically correct, and naturally occurring changes in daylight illumination.

Similar to the work of Delahunt and Brainard [13], one neutral illuminant (CIE $D65$) and four chromatic illuminants (Red, Green, Yellow, Blue) were selected to create images under different light sources. The spectral power distributions of the selected illuminants are shown in figure 2(a) and were created with the use of the CIE daylight basis function, as described in [9]. In figure 2(b), images of scene 3 rendered under these four illuminants are shown.

Real-world data. The second data set consisted of real-world images and were a subset of 50, both indoor and outdoor, images taken from a data set that is widely

Fig. 1. Four examples of the hyperspectral scenes used in this study are shown in figures (a)-(d), rendered under the neutral $D65$ illuminant. In figures (e)-(h), four examples of the real-world scenes are shown.

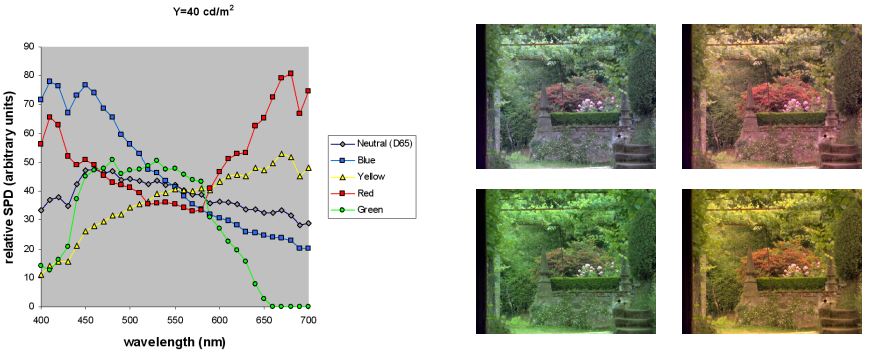
used for performance evaluation of color constancy methods [5]. The original data set consists of over $11,000$ images, and for all images, the ground truth of the color of the light source is known from a grey sphere that was mounted on top of the camera. This grey-sphere was cropped during the experiments. Some example images are shown in figure 1(e)-(h). Images from this data set are not as well calibrated as the previous set, and are therefore mostly used to confirm the results on the hyperspectral data.

4.2 Monitor

Images were viewed on a high-resolution (1600×1200 pixels, 0.27 mm dot pitch) calibrated LCD monitor, an Eizo ColorEdge CG211. The monitor was calibrated to a $D65$ white point of 80 cd/m^2, with gamma 2.2 for each of the three color primaries. CIE 1931 x,y chromaticities coordinates of the primaries were (x,y) = $(0.638, 0.322)$ for red, $(0.299, 0.611)$ for green and $(0.145, 0.058)$ for blue, respectively. These settings closely approximate the sRGB standard monitor profile [15], which was used for rendering the spectral scenes under our illuminants. Spatial uniformity of the display, measured relative to the center of the monitor, was $\Delta\ E_{\mathrm{ab}} < 1.5$ according to the manufacturer's calibration certificates.

4.3 Observers

All observers that participated in the experiments had normal color vision and normal or corrected to normal visual acuity. Subjects were screened for color vision deficiencies with the HRR pseudo-isochromatic plates (4^{th} edition), allowing color vision testing along both the red-green and yellow-blue axes of color space [16]. After taking the color vision test, our subjects first adapted for about five



(a) Illuminant spectra (b) Illuminants applied to scene 3

Fig. 2. Relative spectral power distribution of the illuminants used in the experiments. The illuminants were created with the CIE basis functions for spectral variations in natural daylight, and were scaled such that a perfectly white reflector would have a luminance of 40 cd/m^2. The four chromatic illuminants Red, Green, Yellow and Blue are perceptually at an equal distance (28 ΔE_{ab}) from the neutral (D65) illuminant.

minutes to the light level in a dim room that only received some daylight from a window that was covered with sunscreens (both inside and outside). In the meantime they were made familiar with the experimental procedure.

4.4 Experimental Procedure

The experimental procedure consists of a sequence of image comparisons. The subjects were shown 4 images at once, arranged in a square layout. The images were shown on a gray background having $L^* = 50$ and $a^* = b^* = 0$. The upper two images are (identical) reference images, representing the test scene. The lower two images correspond to the resulting output of two different color constancy algorithms, applied to the original test scene (i.e. the scene under a certain light source). Subjects were instructed to compare the color reproduction of each of the lower images with the upper references. Both the global color impression of the scene and the colors of local image details were to be addressed. Subjects then indicated (by pressing a key on the computer's keyboard) which of the two lower images had the best color reproduction. If the color reproduction of the two test images were identical (as good or as bad), the subjects had the possibility of indicating this. Subjects were told that response time would be measured, but that they were not under time pressure, they could use as much time as they needed to come to a decision.

In each trial of our paired-comparison experiment, two color constancy algorithms are competing, the result of which can be interpreted in terms of a win, a loss or a tie. Each of the five color constancy algorithms is competing with every other algorithm once, for every image and illuminant, in tournament language known as a single round-robin schedule [17]. We applied a scoring mechanism

in which the color constancy algorithm underlying a win was awarded with 1 point and the algorithm underlying a loss with no points. In case of a tie, the competing algorithms both received 0.5 point. Ranking of the algorithms can then be performed by simply comparing the total number of points. The above scoring mechanism is straightforward and makes no distributional assumptions.

5 Results

Experimental results are processed on an "average observer" basis. The inter-observer variability will be analyzed first, after which the results of the observers are averaged to come to robust subjective scores. Next, correlation between these subjective scores and the several objective measures is determined using linear regression. Since the objective measures are absolute error values and the subjective measure depicts a relative relation between the algorithms, the objective measures are converted to relative values. This is done by using the same round-robin schedule as for the human observers, this time using the error values are criterion if one result is better than another.

5.1 Hyperspectral Data

The experiments on the hyperspectral data were run in two sessions, with 4 scenes per session. Per session, a total of 160 comparisons were made (4 scenes $\times$ 4 illuminants $\times$ 10 algorithm combinations). Half of the subjects started with the second set. The two images that were to be compared in a trial always belonged to the same chromatic illuminant. The sequence of the trials was randomized and the two test images were randomly assigned to left and right positions.

Eight observers participated in this experiment, 4 men and 4 women, with ages ranging from 24 to 43 (an average of 34.6). At a viewing distance of about 60 cm, each of the four images subtended a visual angle of $16.6^\circ \times 12.7^\circ$. Horizontal and vertical separation between images was 2.1° and 0.9°, respectively.

Inter-observer variability. As a measure of the inter-observer variability, the individual differences from the mean observer scores are computed, a procedure that is often used in studies involving visual judgements, e.g. [18,19]. For each observer, the correlation coefficient of his/her average algorithm scores (averaged over scenes and illuminants) with the algorithm scores of the average observer is computed. The correlation coefficients so obtained varied from 0.974 to 0.999, with an average of 0.990. Correlation coefficients between scores of the individual observers ranged from 0.937 to 0.997. The significance of this result becomes clear when comparing these high values with the values that are obtained from random data. Based on random generated responses for each trial, with 45%, 45%, 10% chances for a win, loss or tie, respectively, the correlation coefficients of the individual "observers" range from 0.074 to 0.948, with an average of 0.396. Correlation coefficients between individual observers in this case ranged from -0.693 to 0.945. Since the agreement between observers is considered good, in the remainder we will discuss the results only for the average observer.

Mathematical measures vs. subjective scores. First, the angular error d_{angle} is analyzed, since this measure is probably the most widely used performance measure in color constancy research. Overall, the correlation between the angular error and the perception of the human observer is reasonably high, with an average correlation coefficient of 0.895, see table 1(a), where the correlation coefficients on the spectral data set for all distance measures are summarized. Also shown in this table are the results of a paired comparison between the different measures. A Student's t-test (at 95% confidence level) is used to test the null hypothesis that the mean correlation coefficients of two distance measures are equal, against the alternative hypothesis that measure A correlates higher with the human observer than measure B. Comparing every distance measure to with all others, a score is generated representing the number of times the null hypothesis is rejected, i.e. the number of times that the correlation coefficient of the given distance measure is significantly better than the other measures.

By zooming in on individual images, it can be seen that for most images, the correlation is relatively high (correlation coefficient $\rho > 0.95$), while for some images the correlation is somewhat lower, but still acceptable ($\rho > 0.8$). In a few cases, however, the correlation is rather low ($\rho < 0.7$). When observing the results of the images with such a low correlation, the weakness of the angular error becomes apparent. For these images, results of some images are judged worse than indicated by the angular error, meaning that human observers do not agree with the angular error. The angular errors for the corresponding images are similar, but visual inspection of the results show that the estimated illuminants (and hence the resulting images) are far from similar. In conclusion, from a perceptual point-of-view, the direction in which the estimated color of the light source deviates from the ground truth is important. Yet, the angular error, by nature, ignores the direction completely.

The correlation between the Euclidean distance and the human observer is similar to the correlation of the angular error, i.e. $\rho = 0.890$. The other two instantiations of the Minkowski-distance, i.e. the Manhattan distance ($\mathcal{L}_1$) and the Chebychev distance ($\mathcal{L}_\infty$), have a correlation coefficient of $\rho = 0.893$ and $\rho = 0.817$, respectively. The correlation coefficients of other Minkowski-type distance measures are not shown here, but vary between $\rho = 0.89$ and $\rho = 0.82$. In conclusion, none of these mathematical distance measures is significantly different from the others.

Perceptual measures vs. subjective scores. First, the estimated illuminant and the ground truth are converted from normalized-rgb to RGB-values. This is done by computing the two corresponding diagonal mappings to a perfect, white, reflectance, in order to obtain the RGB-values of a perfect reflectance under the two light sources. These RGB-values are then converted to XYZ and the other color spaces, after which they can be compared using any of the mathematical measures. For simplicity, the Euclidean distance is used.

For comparison, recall that the correlation between the human observers and the Euclidean distance of the normalized-rgb values is 0.895. When computing the correlation of the human observers with the Euclidean distance in

different color spaces, the lightness channel L^* is omitted, since the intensity of all estimates is artificially imposed and similar for all light sources. Correlations of human observers and distance measured in the perceptual spaces $L^*a^*b^*$ ($\rho = 0.902$) and $L^*u^*v^*$ ($\rho = 0.872$) are similar to the correlation of the human observers with the Euclidean distance in normalized-rgb space. When computing the Euclidean distance in color spaces like hue and chroma, the correlation is remarkably low; considering both chroma and hue, correlation is 0.646, which is significantly lower than the correlation of other color spaces. Considering chroma or hue alone, correlation drops even further to $\rho = 0.619$ and $\rho = 0.541$, respectively. In conclusion, using perceptual uniform spaces provide similar or lower correlation then rgb.

As was derived from the analysis of the results of the angular error, it can be beneficial to take the direction of a change in color into consideration. In this paper, this property is computed by the perceptual Euclidean distance (PED), by assigning higher weights for different color channels related to human vision (e.g. for Lum the coefficients are $R = 0.3$, $G = 0.59$ and $B = 0.11$). The question remains, however, which weights to use. For this purpose, an exhaustive search has been performed to find the optimal weighting scheme, denoted by $\text{PED}_{\text{hyperspectral}}$ in table 1(a). The weight-combination $(w_R, w_G, w_B) = (0.20, 0.79, 0.01)$ results in the highest correlation ($\rho = 0.963$), but differences with similar weighting combinations are very small such as Luminance $Lum = 0.3R + 0.59G + 0.11B$ which corresponds to the sensitivity of the human visual system. In conclusion, as the human eye is sensitive according to the well-known Lum sensitivity curve, incorporating this property yields a perceptual sound distance measure providing the highest correlation in the experiments on the spectral data.

Color constancy measures vs. subjective scores. The color constancy index makes use of a distance measure as defined by eq. 13, where b is defined as the distance from the estimated light source to the true light source and a is defined as the distance from the true light source to a white reference light. To compute the distance, the angular error in normalized-rgb, and the Euclidean distance in RGB, $L^*a^*b^*$ and $L^*u^*v^*$ are used. From table 1, it can be derived that the highest correlation with the human observers is obtained when measuring the color constancy index with $L^*a^*b^*$ ($\rho = 0.905$). However, differences between other distance measures are small. In conclusion, color constancy index does not correlate better with human observers than the mathematical measures.

The gamut intersection distance measures the distance of the gamuts under the estimated light source and the ground truth. These gamuts are created by applying the corresponding diagonal mappings to a canonical gamut. This canonical gamut is defined as the gamut of all colors under a known, often white, light source and is constructed using a, widely-used, set of 1995 surface spectra [6] combined with a perfect white illuminant. The correlation of this measure is surprisingly high, see table 1: $\rho = 0.965$, which is even slightly higher than the correlation of the Perceptual Euclidean Distance (PED).

Table 1. An overview of the correlation coefficients ρ of several distance measures and using several color spaces, with respect to the human observers. Significance is shown using a Student's t-test (at the 95% confidence level). By comparing every distance measure with all others, a score is generated representing the number of times the null hypothesis (i.e. two distances measures have a similar mean correlation coefficient) is rejected. The results of the experiments on the hyperspectral data are shown in table (a), the results on the real-world data are shown in table (b).

(a) Hyperspectral data

Measure	ρ	T-test (#)
d_{angle}	0.895	3
$\mathcal{L}_1$	0.893	3
$\mathcal{L}_2$	0.890	3
$\mathcal{L}_\infty$	0.817	3
$\mathcal{L}_2 - L^*a^*b^*$	0.902	4
$\mathcal{L}_2 - L^*u^*v^*$	0.872	3
$\mathcal{L}_2 - C + H$	0.646	0
$\mathcal{L}_2 - C$	0.619	0
$\mathcal{L}_2 - H$	0.541	0
$\text{PED}_{\text{hyperspectral}}$	0.963	13
$\mathbf{PED_{proposed}}$	**0.960**	13
$\text{CCI}(d_{\text{angle}})$	0.895	3
$\text{CCI}(\mathcal{L}_{2,RGB})$	0.893	3
$\text{CCI}(\mathcal{L}_{2,L^*a^*b^*})$	0.905	4
$\text{CCI}(\mathcal{L}_{2,L^*u^*v^*})$	0.880	3
d_{gamut}	0.965	13

(b) Real-world data

Measure	ρ	T-test (#)
d_{angle}	0.926	3
$\mathcal{L}_1$	0.930	3
$\mathcal{L}_2$	0.928	3
$\mathcal{L}_\infty$	0.906	3
$\mathcal{L}_2 - L^*a^*b^*$	0.927	3
$\mathcal{L}_2 - L^*u^*v^*$	0.925	3
$\mathcal{L}_2 - C + H$	0.593	1
$\mathcal{L}_2 - C$	0.562	1
$\mathcal{L}_2 - H$	0.348	0
$\text{PED}_{\text{real-world}}$	0.961	14
$\mathbf{PED_{proposed}}$	**0.957**	14
$\text{CCI}(d_{\text{angle}})$	0.931	3
$\text{CCI}(\mathcal{L}_{2,RGB})$	0.929	3
$\text{CCI}(\mathcal{L}_{2,L^*a^*b^*})$	0.921	3
$\text{CCI}(\mathcal{L}_{2,L^*u^*v^*})$	0.927	3
d_{gamut}	0.908	3

Discussion. From table 1(a), it is derived that the correlation of the angular error with the judgment of the human observers is reasonable, and similar to the other mathematical measures, i.e. there is no significant difference at the 95% confidence level. Measuring the distance in perceptual color spaces like $L^*a^*b^*$ and $L^*u^*v^*$ does not increase the correlation with human observers. Using chroma C and hue H significantly decrease the correlation with the human observers. The gamut intersection distance and the perceptual Euclidean distance (PED) have the highest correlation with the human observers. In fact, they have significantly higher (at the 95% confidence level) correlation than all other distance measures. Hence, the gamut and perceptual Euclidean distances are significantly better than all other distance measures on spectral data set.

5.2 Real-World Data

The experiments on the real-world data were run in three sessions, with the number of images equally divided in three parts. The sequence of the sets was randomized for every observer. In this experiment, seven observers participated (4 men and 3 women), with ages ranging from 24 to 43. The difference between the observers was analyzed similarly to the experiments on the hyper-spectral data, and again the agreement of the individual observers was found to be sufficiently high.

Objective vs. subjective scores. In general, the same trends on this data set as on the hyperspectral data are observed, see table 1(b). In general, the correlation coefficients are slightly higher than the spectral data set, but the ordering between the different measures remains the same. For the mathematical measures, the angular distance ($\rho = 0.926$), the Manhattan distance ($\rho = 0.930$) and the Euclidean distance ($\rho = 0.928$) are similar, while the Chebychev distance has a lower correlation with human observers ($\rho = 0.906$). Results of the perceptual measures also show a similar trend. Correlation coefficients of the perceptual color spaces are similar to the mathematical measures, while the intuitive color spaces are significantly lower. Again the perceptual Euclidean distance (PED) has the highest correlation ($\rho = 0.961$). This correlation is obtained with the weights $(w_R, w_G, w_B) = (0.21, 0.71, 0.08)$, denoted $\text{PED}_{\text{real-world}}$ in table 1(b). The results for the color constancy specific distances are slightly different from the results obtained from the hyperspectral data. The results of the color constancy index are similar, but the correlation of the gamut intersection distance with the human observers is considerably lower on this data set.

Discussion. The results of the experiments on the real-world data set, see table 1(b), correspond to the results of the experiments on the hyperspectral data. Note, though, that the images in this data set are gamma-corrected (with an unknown value for gamma) before the color constancy algorithms are used to color correct the images. Applying gamma-correction previously to the color constancy algorithms affects the performance of the used algorithms, but this was not investigated in this paper.

The most noticeable difference between the results on this data set and the results on the previous data set is the correlation of the gamut intersection distance. This distance has the highest correlation with the human observers on the hyperspectral data. However, on the real-world data set, the correlation is considerably lower, though not significant, than the other measures. The correlation of the perceptual Euclidean distance on the real-world data is still significantly higher than the correlation of all other distance measures. To obtain a robust, stable combination of weights, the results of the exhaustive search on the hyperspectral data and the real-world data are averaged. The optimal correlation is found for the weight-combination $(w_R, w_G, w_B) = (0.26, 0.7, 0.04)$, which is the weight-combination we propose to use to compute the PED. Using these weights, correlation of the perceptual Euclidean distance with human observers on the hyperspectral data is 0.960, and on the real-world data is 0.957, denoted PED_{proposed} in table 1(a) and (b), both still significantly higher (at the 95% confidence level) than all other distance measures.

6 Conclusion

In this paper, a taxonomy of different distance measures for color constancy algorithms has been presented. Correlation has been analyzed between the observed quality of the output images and the different distance measures for illuminant estimates. Distance measures have been investigated to what extent they mimic differences in color naturalness of images as obtained by humans.

Based on the theoretical and experimental results on spectral and real-world data sets, it can be concluded that the perceptual Euclidean distance (PED) with weight-coefficients ($w_R = 0.26$, $w_G = 0.70$, $w_B = 0.04$) finds its roots in human vision and correlates significantly higher than all other distance measures including the angular error and Euclidean distance.

References

1. van de Weijer, J., Gevers, T., Gijsenij, A.: Edge-based color constancy. IEEE Transactions on Image Processing 16(9), 2207–2214 (2007)
2. Land, E.: The retinex theory of color vision. Scientific American 237(6), 108–128 (1977)
3. Buchsbaum, G.: A spatial processor model for object colour perception. J. Franklin Institute 310(1), 1–26 (1980)
4. Finlayson, G., Trezzi, E.: Shades of gray and colour constancy. In: Proc. CIC, pp. 37–41 (2004)
5. Ciurea, F., Funt, B.: A large image database for color constancy research. In: Proc. CIC, pp. 160–164 (2003)
6. Barnard, K., Martin, L., Funt, B., Coath, A.: A data set for color research. Color Research and Application 27(3), 147–151 (2002)
7. Hordley, S., Finlayson, G.: Reevaluation of color constancy algorithm performance. J. Opt. Soc. America A 23(5), 1008–1020 (2006)
8. von Kries, J.: Influence of adaptation on the effects produced by luminous stimuli. In: MacAdam, D. (ed.) Sources of Color Vision, pp. 109–119. MIT Press, Cambridge (1970)
9. Wyszecki, G., Stiles, W.: Color science: concepts and methods, quantitative data and formulae. John Wiley & sons, Chichester (2000)
10. Slater, J.: Modern television systems to HDTV and beyond. Taylor & Francis Group, Abington (2004)
11. Arend, L., Reeves, A., Schirillo, J., Goldstein, R.: Simultaneous color constancy: papers with diverse munsell values. J. Opt. Soc. America A 8(4), 661–672 (1991)
12. Brunswik, E.: Zur entwicklung der albedowahrnehmung. Zeitschrift fur Psychologie 109, 40–115 (1928)
13. Delahunt, P., Brainard, D.: Does human color constancy incorporate the statistical regularity of natural daylight? Journal of Vision 4(2), 57–81 (2004)
14. Foster, D., Nascimento, S., Amano, K.: Information limits on neural identification of colored surfaces in natural scenes. Visual Neuroscience 21, 331–336 (2004)
15. Stokes, M., Anderson, M., Chandrasekar, S., Motta, R.: A standard default color space for the internet-srgb (1996), `www.w3.org/Graphics/Color/sRGB.html`
16. Bailey, J., Neitz, M., Tait, D., Neitz, J.: Evaluation of an updated hrr color vision test. Visual Neuroscience 22, 431–436 (2004)
17. David, H.: Ranking from unbalanced paired-comparison data. Biometrika 74, 432–436 (1987)
18. Alfvin, R., Fairchild, M.: Observer variability in metameric color matches using color reproduction media. Color Research and Application 22, 174–188 (1997)
19. Kirchner, E., van den Kieboom, G., Njo, L., Supr, R., Gottenbos, R.: Observation of visual texture of metallic and pearlescent materials. Color Research and Application 32, 256–266 (2007)

Scale Invariant Action Recognition Using Compound Features Mined from Dense Spatio-temporal Corners

Andrew Gilbert, John Illingworth, and Richard Bowden

CVSSP, University of Surrey, Guildford,
GU2 7XH, England

Abstract. The use of sparse invariant features to recognise classes of actions or objects has become common in the literature. However, features are often "engineered" to be both sparse and invariant to transformation and it is assumed that they provide the greatest discriminative information. To tackle activity recognition, we propose learning compound features that are assembled from simple 2D corners in both space and time. Each corner is encoded in relation to its neighbours and from an over complete set (in excess of 1 million possible features), compound features are extracted using data mining. The final classifier, consisting of sets of compound features, can then be applied to recognise and localise an activity in real-time while providing superior performance to other state-of-the-art approaches (including those based upon sparse feature detectors). Furthermore, the approach requires only weak supervision in the form of class labels for each training sequence. No ground truth position or temporal alignment is required during training.

1 Introduction

The recognition of human activity within a video sequence is a popular problem. It is a difficult as subjects can vary in size, appearance and pose. Furthermore, cluttered backgrounds and occlusion can also cause methods to fail. Varying illumination and incorrect temporal alignment of actions can cause large within (intra) class variation. While inter-class variation can be low due to similarity in motion and appearance. To illustrate, Figure 3(d), (e) & (f) show example frames from the KTH [1] dataset for the categories 'jogging', 'running' and 'walking' respectively. Scaling issues aside, the similarity of these static frames illustrates the need to use temporal information when identifying actions.

Within the object recognition community, learning strategies for feature selection have proven themselves successful at building classifiers from large sets of possible features e.g. Boosting [2]. Although similar approaches have been applied to the spatio-temporal activity domain [3] [4], such approaches do not scale well due to the number of features and also issues with time alignment/scaling. Therefore sparse, but more complex, feature descriptors have been proposed [5] [1] [6]. The sparsity of such features makes the problem of recognition tractable but such sparsity also means potential information is lost to the recognition architecture.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 222–233, 2008.

Our approach is based upon extracting very low-level features (corners in xy, xt and yt) from videos and combining them locally to form high-level, compound, spatio-temporal features. The method outlined in this paper takes advantage of data mining to assemble the compound features using an *association rule* data mining technique [7] which efficiently discovers frequently reoccurring combinations/rules. The resulting rules are then used to form a classifier which provides a likelihood of the occurrence and position of an action in a sequence.

Association rule data mining was recently employed by Quack *et al.* [8] to group SIFT descriptors for object recognition. We use the algorithm in a similar fashion but instead of using it to group high-level features, we use it to build high-level compound features from a noisy and over-complete set of low-level spatio and spatio-temporal features (corners). This is then applied to the task of activity recognition. We compare encoding only relative spatial offsets, which provides scale invariance, to the spatial grid proposed by Quack *et al.* and demonstrate that, due to increased scale invariance, higher performance is achieved. Learning is performed with only sequence class labels rather than full spatio-temporal segmentation. The resulting classifier is capable of both recognising and localising activities in video. Furthermore, we demonstrate that efficient matching can be used to obtain real-time action recognition on video sequences.

2 Related Work

Within object recognition, the use of spatial information of local features has shown considerable success [8] [9] [10]. Many action recognition methods also use a sparse selection of local interest points. Schüldt *et al.* [1] and Dollar *et al.* [5] employ sparse spatio-temporal features for the recognition of human (and mice) actions. Schüldt takes a codebook and bag-of-words approach applied to single images to produce a histogram of informative words or features for each action. Niebles and Fei-Fei [11] use a hierarchical model that can be characterized as a constellation of bags-of-words. Similarly Dollar take the bag-of-words approach but argue for an even sparser sampling of the interest points. This improves the performance on the same video sets. However, with such a sparse set of points, the choice of feature used is important. Scovanner *et al.* [12] extended the 2D SIFT descriptor [13] into three dimensions, by adding a further dimension to the orientation histogram. This encodes temporal information and dramatically outperforms the 2D version. To model motion between frames, optical flow [14] [15] can be applied as was used by Laptev [6] in addition to a shape model to detect drinking and smoking actions. Yang Song *et al.* [16] use a triangular lattice of grouped point features to encode layout.

There are relatively few examples of mining applied to the imaging domain. Tesic *et al.* [17] use a Data mining approach to find the spatial associations between classes of texture from aerial photos. Similarly Ding *et al.* [18] derive association rules on Remote Sensed Imagery data using a Peano Count Tree (P-tree) structure with an extension of the more common *APriori* [7] algorithm.

Chum *et al.* [19] used data mining to find near duplicate images within a database of photographs. Our approach uses data mining as a feature selection process for activity recognition.

3 Building Compound Features

3.1 Extracting Temporal Harris Interest Points

In contrast to very sparse feature detectors, we build our detection system upon corner features. The rationale for using corners are they are simple to compute, largely invariant to both lighting and geometric transformation, and provide an over-complete feature set from which to build more complex compound features. To identify and locate the interest points in images, the well known Harris corner detector [20] is applied in (x, y), (x, t) and (y, t) as a 3x3 patch. Unlike the 3D corners of [6], which are sparse, detecting 2D corners in 3 planes produces a relatively large and over complete set of features, with typically 400 corners detected per frame on the KTH data. Each corner feature has a dominant gradient orientation, this orientation can be used to encode the feature type into one of a set of discrete corner orientations. Figure 1 shows the example corner detections on two frames. It shows that in 1(a), most features occur around the hands espe-

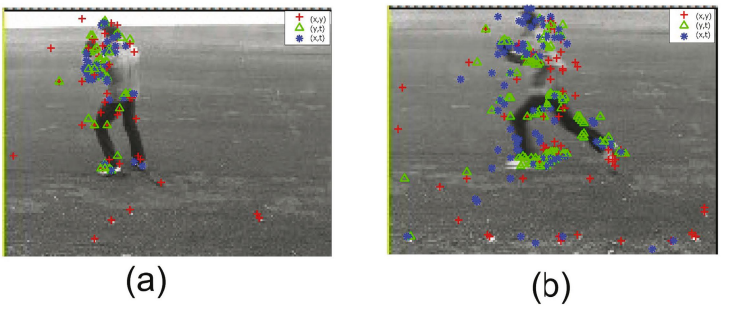


Fig. 1. Corner Detection on two Frames, (a) A Boxing Sequence, (b) A Running Sequence

cially in the (x,t) and (y,t) dimensions. A similar pattern occurs in 1(b) with a large amount of features around the feet, hands and head. The large number of features detected make clustering methods for code book construction unsuitable but the simplicity of the features also makes such an approach redundant.

In order to overcome the effects of scale, the interest point detector was applied to the video sequences across scale space to detect corners at different scales [21]. This was achieved by successively 2x2 block averaging the image frames. Table 1 shows the scale, image size and effective interest point patch sizes. Each feature is now encoded by a 3 digit vector (s, c, o). The encoding includes the scale $s \in \{1, ..., 4\}$ corresponding to the interest point size $\{3x3, ..., 48x48\}$, $c \in \{1, ..., 3\}$ the channel the interest point was detected in $\{xy, xt, yt\}$ and the

Table 1. Table showing the image and relative interest point patch sizes

Scale	1	2	3	4
Image Size	160x120	80x60	40x30	20x15
Interest Point Size	3x3	6x6	24x24	48x48

gradient orientation of the interest point $o \in \{1, ..., n\}$. Orientation is quantised into n discrete orientations. In our experiments $n = 8$ so orientation is quantised into 45^o bins aligned with a points of a compass. Figure 2(a) shows an example of the vector encoding.

3.2 Spatial Grouping

The spatial configuration of features is key to object recognition and has been demonstrated to significantly enhance action recognition when modelled independently from temporal information [6]. Quack *et al.* [8] encoded the spatial layout of features by quantising the space around a feature into a grid and assigning features to one of those locations. Where, the size of the grid is dependant on the scale of the detected SIFT feature to provide robustness to scale. This approach is difficult to achieve for less descriptive interest points such as corners, so our approach is to define neighbourhoods centred upon the feature that encode the relative displacement in terms of angle rather than distance hence achieving scale invariance. To do this, each detected interest point forms the centre of a neighbourhood. The neighbourhood is divided into 8 quadrants in the x, y, t domain which radiate from the centre of the neighbourhood out to the borders of the image in x, y and one frame either side either $t - 1$ or $t, t+1$ (see Figure2(b-c)). Each quadrant is given a label, all feature codes found

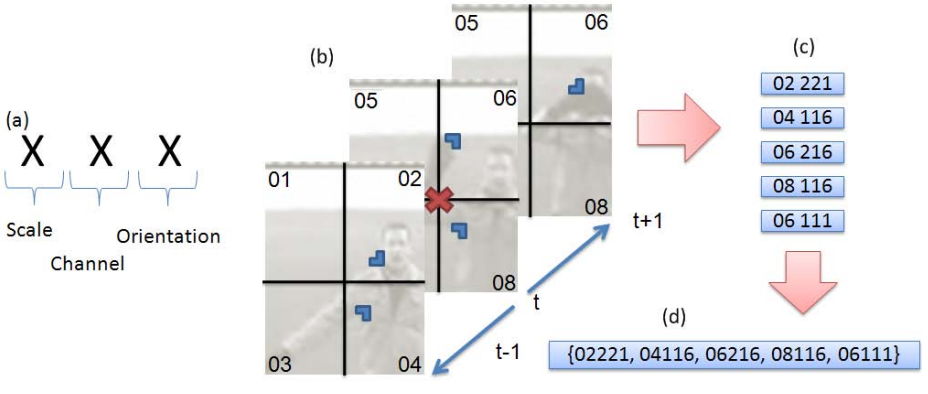


Fig. 2. (a) The three parts that make up a local feature descriptor. (b) shows a close-up example of a 2x2x2 neighbourhood of an interest point, with five local features shown as corners. (c) shows the spatial and temporal encoding applied to each local feature. (d) Concatenating the local features into a transaction vector for this interest point.

within a unique quadrant are appended with the quadrant label. A vector of these elements is formed for every interest point found in the video sequence and contains the relative spatial encoding to all other features on the frame. For efficiency this is done by using a look-up to an integral histogram of the 3 digit feature codes. This newly formed set is called a transaction set, T, where the spatially encoded features contained within it are items. To summarise, Figure 2 shows the formation of a single transaction set, from five individual local features.

For each interest point a transaction set is formed. These are collected together to compute a transaction database for each action. For a typical example video from the KTH dataset, this database contains around 500,000 transactions for each action, where a single transaction contains around 400 items. To condense or summarise this vast amount of information, data mining is employed.

4 Data Mining

Association rule [22] mining was originally developed for the analysis of customers supermarket baskets. Its purpose, to find regularity in the shopping behaviour of customers, by finding association rules within millions of transactions. An association rule is a relationship of the form $\mathbf{A} \Rightarrow \mathbf{C}$, where $\mathbf{A}$ and $\mathbf{C}$ are itemsets. $\mathbf{A}$ is called the antecedent and $\mathbf{C}$ the consequence. An example of the rule can be, customers who purchase an item in $\mathbf{A}$ are very likely to purchase another item in $\mathbf{C}$ at the same time. As there will be billions of transactions and therefore millions of possible association rules, efficient algorithms have been developed to quickly formulate the rules. One such algorithm is the popular *APriori* algorithm developed by Agrawal [7].

4.1 Frequent Itemsets

It can be said that transaction T *supports* an itemset $\mathbf{A}$ if $\mathbf{A} \subseteq T$. The algorithm attempts to find subsets which are frequent to at least a minimum number T_{Conf} (confidence threshold) of the items. If $\{\mathbf{A}, \mathbf{B}\}$ is a frequent itemset, both subsets $\mathbf{A}$ and $\mathbf{B}$ must be frequent itemsets as well. This fact is exploited by the algorithm to increase efficiency. *APriori* uses a "bottom up" approach, where frequent subsets are extended one item at a time, and groups of candidates are tested against the confidence threshold.

4.2 Association Rules

The association rule is the expression $\{\mathbf{A}, \mathbf{B}\} \Rightarrow \mathbf{C}$ where given itemsets $\mathbf{A}$ and $\mathbf{B}$, the itemset $\mathbf{C}$ will frequently occur. The belief of each rule is measured by a support and confidence value.

Support Rule. The support, $sup(\{\mathbf{A}, \mathbf{B}\} \Rightarrow \mathbf{C})$ of a rule, measures the statistical significance of a rule, the probability that a transaction contains itemsets $\mathbf{A}$ and $\mathbf{B}$.

Confidence Rule. The confidence rule is used to evaluate an association rule. The confidence of a rule $Conf(\{\mathbf{A}, \mathbf{B}\} \Rightarrow \mathbf{C})$ is the support of the set of all items that appear in the rule, divided by the support of the antecedent of the rule. This means the confidence of a rule is the number of times in which the rule is correct relative to the number of cases in which it is applicable. This measure is used to select association rules, if it's confidence exceeds a threshold T_{Conf}.

4.3 Mining for Frequent and Distinctive Itemsets

Once the local feature neighbourhoods are formed into transactions, the frequent and distinctive itemsets that make up the transactions must be found. This is achieved by running the APriori [7] algorithm on the transaction database, to find the frequently occurring itemset configurations. It is important that the resulting frequent itemsets are distinctive inter class. Therefore positive examples of an action transaction were appended with a 1. While an equal sub set of all other actions are appended with a 0 to provide the negative examples for training. This is used as it is important the resulting mined itemset configurations are only frequent in assigning a feature to an action. Given an association rule **AS**, its confidence is used to look for rules that have a high probability of being correct. Meaning that a chosen frequent itemset must imply the specific action, as shown in Equation 1.

$$Conf(\mathbf{AS} \Rightarrow action) > T_{Conf} \tag{1}$$

The mining algorithm allows for the efficient computation of frequent itemset configurations. In our experiments, a transaction file consists of over 1 million possible transactions with each individual transaction containing around 400 items. This size would prohibit many semi-unsupervised learning methods. However the efficient approach of the APriori algorithm, allows for the frequent itemsets to be found within 1 hour, on standard desktop PC. Once completed, each association rule, **AS**, which satisfies equation 1 is added to a Frequent Mined Configuration vector $\mathbf{M}$. Where $\mathbf{M} = \{\mathbf{AS}_1, ..., \mathbf{AS}_N\}$ for the N association rules.

5 Classifying Actions

The Frequent Mined Configurations $\mathbf{M}$ for a specific action represents the frequent and distinctive itemsets of the training action sequences. Given a new query action sequence, the same feature extraction and spatial grouping of section 3 is applied to the query video. This forms a new query set of transactions $\mathbf{D}_{query} = \{T_1, ..., T_n\}$. To classify the action, a global classifier is used. However, in practice the extraction process is not required as the transaction rules can be applied as a lookup to the integral histogram.

5.1 Global Classifier

As shown in equation 2, the global classifier exhaustively compares a specific action (α) itemset $\mathbf{M}_\alpha$ and the image feature combinations in the transaction set

$\mathbf{D}_{query}$ for a triplet of frames $\mathbf{F} = \{f_{t-1}, f_t, f_{t+1}\}$ within a test sequence. It works as a voting scheme by accumulating the occurrences of the mined compound features.

$$Conf_\alpha(\mathbf{F}) = \frac{1}{N_\alpha * n} \sum_{\forall \mathbf{D}_{query}} m(T_i, \mathbf{M}_\alpha) \tag{2}$$

where N_α is the number of transaction sets mined from the training data, and n is the number of transactions or neighbourhoods in the current time step. $m(T_i, \mathbf{M}_\alpha)$ describes if a transaction is present in the mined configuration.

$$m(T_i, \mathbf{M}_\alpha) = \begin{cases} Conf(T_i \Rightarrow \alpha) & T_i \in \mathbf{M}_\alpha \\ 0 & otherwise \end{cases} \tag{3}$$

This is repeated over the complete test sequence of an action with all the mined action configurations to find the likelihood of the sequence. A correct match will occur often in equation 3 as the mining will select frequently reoccurring items that are distinct to other actions. The use of a codebook allows the classifier to run at approximately 12fps on unoptimised C++ code on a standard pc. Each video sequence is then classified as the action, α, for which the votes are maximised.

5.2 Action Localisation

As each transaction encodes the relative location of features into one of eight quadrants. Each transaction found can vote for which of the eight quadrants other features should be located in. A comparison is made between the features in the transaction set $\mathbf{D}_{query}$, with the Frequent Mined Configuration vector features $\mathbf{M}$. If a match is found, all pixels within a quadrant are incremented by 1 on a likelihood image. This is repeated for all matched features, eventually causing the likelihood image to produce a peak around the centre of the action. An example of this is shown in Figure 6(f), where Figure 6(e) shows the thresholded centre of the action.

6 Experiments

To evaluate the approach, two sets of videos were used. The KTH human action dataset of Schüldt *et al.* [1] is a popular dataset for action recognition, containing 6 different actions; boxing, hand-waving, hand-clapping, jogging, running and walking. There are a total of 25 people performing each action 4 times, giving 599 videos, (1 is missing) totalling 2396 unique actions. The portion of data for training and testing was identical to that proposed by Schüldt [1] to allow direct comparison of results. In order to demonstrate localisation in the presence of multiple subjects, a sequence consisting of a two people walking through the scene, with one person stopping to perform a single hand wave was recorded. Examples of the two sequences are shown in Figure 3. The sequences have different scales, and temporal speeds of actions, and some of the action classes

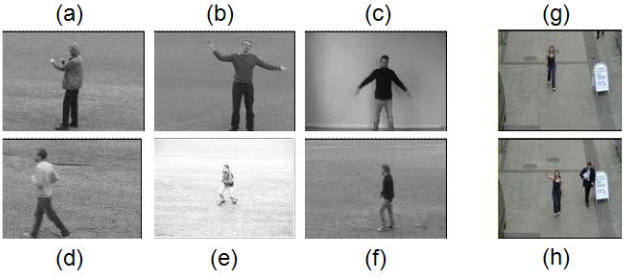


Fig. 3. Example frames from the two datasets, (a-f) KTH, (g,h) multi-person dataset: (a) boxing, (b) hand-clapping, (c) hand-waving, (d) jogging, (e) running, (f) walking, (g) one person walking, (h) one person walking, one person hand-waving

have very similar appearances. The training sequences, were used to produce a Frequent Mined Configuration vector M for each of the six actions containing up to 10 compound features in length. These were then used to classify each of the test sequences. Figure 4(a) shows the classification confusion matrix using the scale invariant grid approach proposed within this paper, where good class separability is exhibited. The results show relatively little confusion compared to other approaches with minor confusion between boxing and clapping. Jogging and running also causes some confusion but, this is consistent with previous approaches. This confusion is due to the inherent similarity of the motion. In Figure 4(b) the experiments were repeated using a *fixed size* 4x4 grid similar to [8]. To investigate the importance of the spatial and temporal compounding of individual features, Figure 5 shows the effect on overall accuracy (left axis) as the minimum item size in the transaction sets is increased. It can be seen

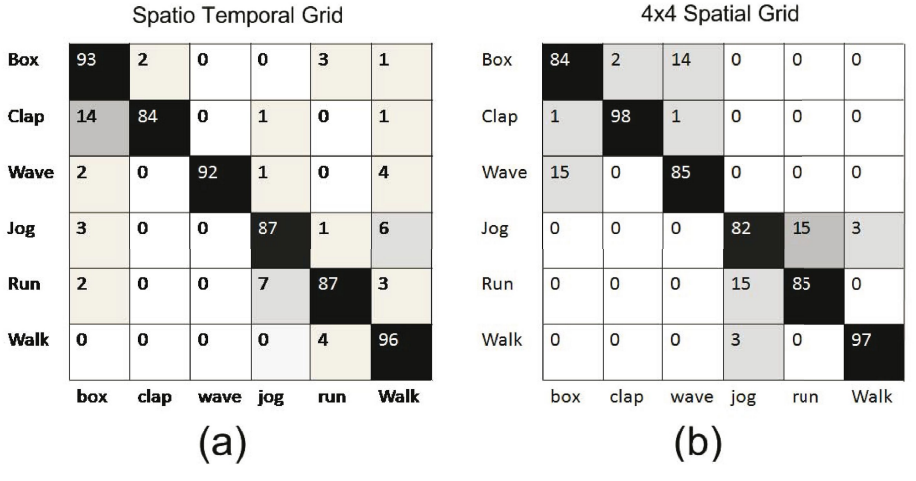


Fig. 4. (a)The confusion matrix of the Data Mined corner descriptor on the KTH dataset with **Scale Invariance**. (b) The confusion matrix of the Data Mined corner descriptor on the KTH dataset with a fixed non scale invariant spatial grouping.

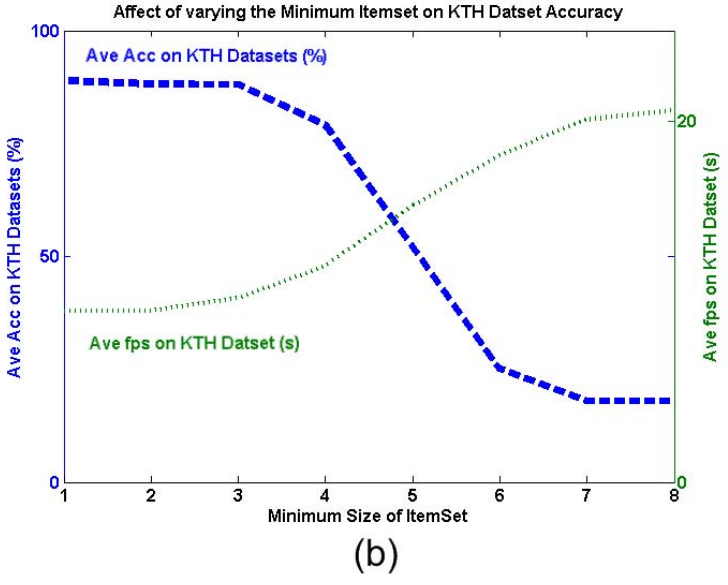


Fig. 5. The classification accuracy as the itemset size is increased

that no drop in performance is found in discarding itemsets under fours features in size. This confirms the importance of the grouping of the single features together. Disregarding these features gives an increase in frame rate from 9.5fps to 12fps, due to the reduced feature complexity. Therefore the small feature groups can be discarded with no loss in accuracy to further increase speed.

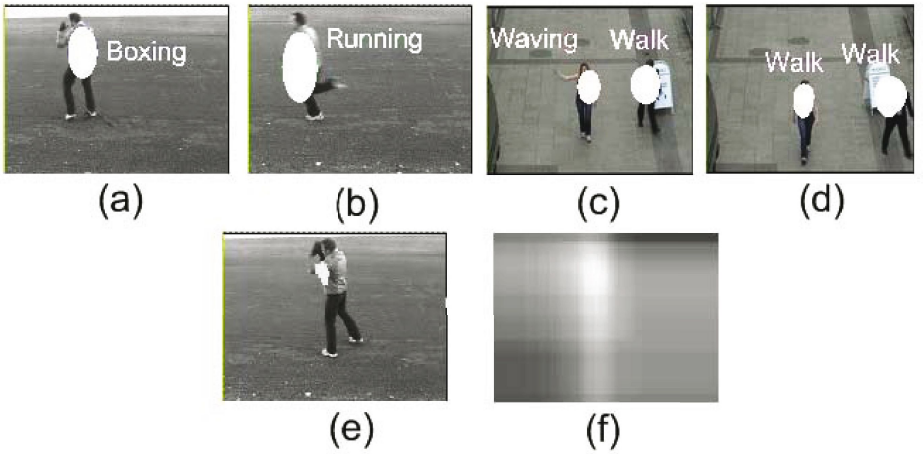


Fig. 6. (a)Localised boxing action (b) Localised running action Likelihood image, (c) Multiple localised waving and walking actions, (d) Multiple localised walking actions (e) Thresholded localised action (f) Localisation likelihood image for image (e).

The classification can also be used to localise and identify multiple actions in frames. Figure 6 shows the localisation of four frames actions. Two are from the KTH sequences (a) and (b), while (c) and (d) are from the multi-person outdoor sequence, it contains two people, walking where one stops and waves. In addition the wave action is much less exaggerated than the KTH version and only single handed. Despite these constraints, as shown in Figure 6(c) and (d), the actions are correctly localised and identified.

Table 2 shows results by a number of previous published works on the same dataset, including **Spat-Temp Dollar:** The very sparse spatio-temporal descriptor by Dollar [5] and **Subseq Boost Nowozin:** The boosted SVM classifier by Nowozin [23]. As Table 6 shows, our proposed technique, **Scale Invariant**

Table 2. Comparison of Average precision compared to other techniques on KTH action recognition Dataset

Method	Average Precision
Nowozin *et al.* [23] Subseq Boost SVM	87.04%
Wong and Cipolla [24] Subspace SVM	86.60%
Niebles *et al.* [25] pLSA model	81.50%
Dollar *et al.* [5] Spat-Temp	81.20%
Schüldt *et al.* [1] SVM Split	71.71%
Ke *et al.* [3] Vol Boost	62.97%
Fixed Grid Mined Dense Corners	88.50%
Scale Invariant Mined Dense Corners	89.92%

Mined Dense Corners has a higher classification accuracy than other published methods. This is because of the ability of the technique to select optimal low level features for discrimative classification.

7 Conclusion

This paper has presented a method to efficiently learn informative and descriptive local features of actions performed by humans at multiple scales and temporal speeds. Very coarse corner descriptors are grouped spatially to form an over complete set of feature sets that encode local feature layout. The frequently reoccurring features are then learnt in a weakly-supervised approach where only class labels are required using a data mining algorithm. When tested on the popular KTH dataset, impressive results are obtained which outperform other state-of-the-art approaches while maintaining real-time operation (12fps) in an unoptimised implementation. Although no object segmentation is required during training. The final classifiers can be used to perform activity localisation as well as classification.

Acknowledgments

This work is supported by the EU FP6 Project URUS, and the EU FP7 Project DIPLECS.

References

1. Schuldt, C., Laptev, I., Caputo, B.: Recognizing Human Actions: a Local SVM Approach. In: Proc. of International Conference on Pattern Recognition (ICPR 2004), vol. III, pp. 32–36 (2004)
2. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. In: Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2001), vol. I, pp. 511–518 (2001)
3. Ke, Y., Sukthankar, R., Hebert, M.: Efficient Visual Event Detection using Volumetric Features. In: Proc. of IEEE International Conference on Computer Vision (ICCV 2005) (2005)
4. Cooper, H.M., Bowden, R.: Sign Language Recognition Using Boosted Volumetric Features. In: Proc. IAPR Conf. on Machine Vision Applications, pp. 359–362 (2007)
5. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior Recognition via Sparse Spatio-temporal Features. In: ICCCN 2005: Proceedings of the 14th International Conference on Computer Communications and Networks, pp. 65–72 (2005)
6. Laptev, I., Pérez.: Retrieving Actions in Movies. In: Proc. of IEEE International Conference on Computer Vision (ICCV 2007) (2007)
7. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: VLDB 1994, Proceedings of 20th International Conference on Very Large Data Bases, pp. 487–499 (1994)
8. Quack, T., Ferrari, V., Leibe, B., Gool, L.: Efficient Mining of Frequent and Distinctive Feature Configurations. In: Proc. of IEEE International Conference on Computer Vision (ICCV 2007) (2007)
9. Lazebnik, S., Schmid, C., Ponce, J.: Semi-Local Affine Parts for Object Recognition. In: Proc. of BMVA British Machine Vision Conference (BMVC 2004), vol. II, pp. 959–968 (2004)
10. Sivic, J., Zisserman, A.: Video Data Mining using Configurations of Viewpoint Invariant Regions. In: Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2004), vol. I, pp. 488–495 (2004)
11. Niebles, J.C., Fei-Fei, L.: A Hierarchical Model of Shape and Appearance for Human Action Classification. In: Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2007) (2007)
12. Scovanner, P., Ali, S., Shah, M.: A 3-dimensional sift descriptor and its application to action recognition. In: Proc. of MULTIMEDIA 2007, pp. 357–360 (2007)
13. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 20, 91–110 (2003)
14. Dalal, N., Triggs, B., Schmid, C.: Human Detection using Oriented Histograms of Flow and Apperance. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 428–441. Springer, Heidelberg (2006)
15. Lucas, B., Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In: Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI), pp. 674–679 (1998)

16. Song, Y., Goncalves, L., Perona, P.: Unsupervised Learning of Human Motion. Transactions on Pattern Analysis and Machine Intelligence 25, 814–827 (2003)
17. Tesic, J., Newsam, S., Manjunath, B.S.: Mining image datasets using perceptual association rules. In: Proc. SIAM International Conference on Data Mining, Workshop on Mining Scientific and Engineering Datasets, pp. 71–77 (2003)
18. Ding, Q., Ding, Q., Perrizo, W.: Association rule mining on remotely sensed images using p-trees. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 66–79 (2002)
19. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval. In: Proc. IEEE International Conference on Computer Vision (ICCV 2007), pp. 1–8 (2007)
20. Harris, C., Stphens, M.: A Combined Corner and Edge Detector. In: Proc. of Alvey Vision Conference, 189–192 (1988)
21. Fleuret, F., Geman, D.: Coarse to Fine Face Detection. International Journal of Computer Vision 41, 85–107 (2001)
22. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proc. of the 1993 ACM SIGMOD International Conference on Management of Data SIGMOD 1993, pp. 207–216 (1993)
23. Nowozin, S., Bakir, G., Tsuda, K.: Discriminative Subsequence Mining for Action Classification. In: Proc. of IEEE International Conference on Computer Vision (ICCV 2007), pp. 1919–1923 (2007)
24. Wong, S.F., Cipolla, R.: Extracting Spatio Temporal Interest Points using Global Information. In: Proc. of IEEE International Conference on Computer Vision (ICCV 2007) (2007)
25. Niebles, J., Wang, H., Fei-Fei, L.: Unsupervised Learning of Human Action Categories using Spatial-Temporal Words. In: Proc. of BMVA British Machine Vision Conference (BMVC 2006), vol. III, pp. 1249–1259 (2006)

Semi-supervised On-Line Boosting for Robust Tracking*

Helmut Grabner[1,2], Christian Leistner[1], and Horst Bischof[1]

[1] Institute for Computer Graphics and Vision, Graz University of Technology, Austria
{hgrabner,leistner,bischof}@icg.tugraz.at
[2] Computer Vision Laboratory, ETH Zurich, Switzerland
grabner@vision.ee.ethz.ch

Abstract. Recently, on-line adaptation of binary classifiers for tracking have been investigated. On-line learning allows for simple classifiers since only the current view of the object from its surrounding background needs to be discriminiated. However, on-line adaption faces one key problem: Each update of the tracker may introduce an error which, finally, can lead to tracking failure (drifting). The contribution of this paper is a novel on-line semi-supervised boosting method which significantly alleviates the drifting problem in tracking applications. This allows to limit the drifting problem while still staying adaptive to appearance changes. The main idea is to formulate the update process in a semi-supervised fashion as combined decision of a given prior and an on-line classifier. This comes without any parameter tuning. In the experiments, we demonstrate real-time tracking of our SemiBoost tracker on several challenging test sequences where our tracker outperforms other on-line tracking methods.

1 Introduction

Designing robust tracking methods is still an open issue, especially considering various complicated variations that may occur in natural scenes, *e.g.*, shape and appearance changes of the object, illumination variations, partial occlusions of the object, cluttered scenes, *etc.* Recently tracking has been formulated as a classification problem, *i.e.*, the task of tracking is to optimally separate in each frame the object from the background (*e.g.*, Avidan [1] used support vector machines). Also, feature based tracking methods are formulated as classification tasks, *i.e.*, the work of Lepetit et al. [2] uses randomized trees and ferns based on pixel pairs [3] to discriminate key points by classifiers. In these approaches, the object to be tracked is trained a priori. The main motivation for using classifiers in these approaches is the increased speed, *i.e.*, the time is spent at the training stage and a fast classifier is available at the tracking stage. All these approaches use off-line training,

* This work has been supported by the Austrian Joint Research Project Cognitive Vision under projects S9103-N04 and S9104-N04, the FFG project EVis (813399) under the FIT-IT program and the Austrian Science Fund (FWF) under the doctoral program Confluence of Vision and Graphics W1209.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 234–247, 2008.

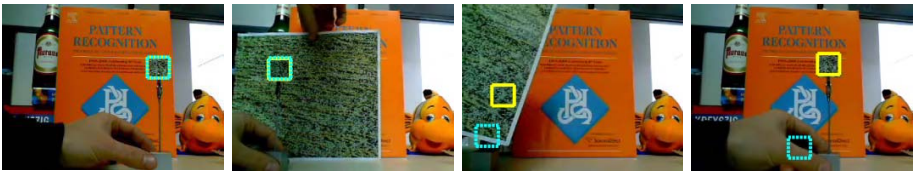


Fig. 1. Tracking of a textured patch with difficult background (same texture). As soon as the object gets occluded the original tracker from [4] (dotted cyan), drifts away. Our proposed SemiBoost tracker (yellow) successfully re-detects the object and continues tracking without drifting.

which has two important limitations. First, all appearance variations need to be covered in advance which implies that the object to be tracked needs to be known beforehand. Tracking will fail if a variation of the object is not covered in the training phase. Second, since the tracker is fixed it has to cope with all different backgrounds, therefore the classifiers are usually quite complex.

In order to cope with these problems the tracker needs to be adaptive. Collins and Liu [5] were among the first to emphasize (and exploit) this principle in a tracker. They proposed a method to adaptively select color features that best discriminate the object from the current background. There has also been considerable work along these lines, *e.g.*, Lim *et al.* [6] used incremental subspace learning for tracker updating and Avidan [7] use an adaptive ensemble of classifiers. Furthermore, Grabner *et al.* [8] have designed an on-line boosting classifier that selects features to discriminate the object from the background. This work has demonstrated that by using on-line feature selection the tracking problem can considerably be simplified and therefore the classifiers can be quite compact and fast. For instance, Fig. 1 depicts a challenging tracking sequence, where a small textured patch is tracked using on-line boosting. Since the trackers are trained to optimally handle foreground/background discrimination, they can handle also such difficult situations where the same texture is used as background. However, when we occlude the object it is lost (since it is no longer visible) and the tracker (continuously updating its representation) starts tracking something different.

Hence, using on-line adaptation we face drifting as the key problem. Each time we make an update to our tracker an error might be introduced, resulting in a tracking error, which may accumulate over time resulting in tracking failures. Matthews *et al.* [9] have pinpointed this problem and proposed a partial solution for template trackers. Looking at this problem from a classification point of view we have the necessity to introduce a "teacher" to train the classifier. Other approaches used a geometric model (*e.g.*, homography for planar objects) for verification [10] and performed updating only when the geometric model is verified. This alleviates the drifting problem but is not applicable in all situations. Co-learning (using multiple trackers operating on different features that train each other) is another strategy proposed in [11]. Combinations of generative and discriminative models are used [12]. Both approaches alleviate

Fig. 2. Detection and tracking can be viewed as the same problem, depending on how fast the classifier adapts to the current scene. On the one side a general object detector (*e.g.*, [14]) is located and on the other side a highly adaptive tracker (*e.g.*, [4]). Our approach is somewhere in between, benefiting from both approaches: (i) be sufficiently adaptive to appearance and illumination changes and (ii) limit (avoid large) drifting by keeping prior information about the object.

the drifting problem to a certain extend but cannot avoid it. Summarizing, we can either use fixed classifiers which per definition do not suffer from the drifting problem, but have limited adaptation capabilities or we can use on-line adaptation and then have to face the drifting problem[1]. In fact, this is not a binary choice as depicted in Fig. 2.

In this paper, we explore the continuum between a fixed detector and on-line learning methods as depicted in Fig. 2. Recently, this has also been investigated by Li *et al.* [15] for tracking in low-frame rates. However, in order to formulate this problem in a principled manner we use ideas from semi-supervised learning (see [16] for a recent survey). In particular, we use the recently proposed SemiBoost [17,18] for learning a classifier. Labeled data (or a previously trained model) is used as a prior and the data collected during tracking as unlabeled samples. This allows us to formulate the tracker update problem in a natural manner. Additionally, this solves the problem of how to weight the a priori information and the on-line classifier without parameter tuning. The major contribution is an on-line formulation of semi-supervised boosting which is a requirement for using this algorithm for tracking.

Back to our example shown in Fig. 1. The proposed approach performs similar to the former on-line tracker up to the third subfigure, where both get lost. Yet, in contrast to the on-line boosting, as soon as the object becomes visible again it is re-detected by the SemiBoost tracker (using the a priori knowledge) while the on-line boosted tracker meanwhile has adapted itself to a completely different region which it finally tries to track.

The reminder of the paper is organized as follows. Section 2 shortly reviews on-line boosting for feature selection and a recently published variant of semi-supervised boosting called SemiBoost[18]. In Section 3, we present our novel on-line SemiBoosting method, which is then used in a tracking application shown in Section 4. Section 5 presents some detailed experiments and results. Finally, our work concludes with Section 6.

[1] In fact, this is another instance of the stability plasticity dilemma [13].

2 Preliminaries

2.1 Off-Line Boosting for Feature Selection

Boosting is a widely [19] used technique in machine learning for improving the accuracy of any given learning algorithm. In this work, we focus on the (discrete) AdaBoost algorithm, which has been introduced by Freund and Shapire [20]. The algorithm can be summarized as follows: given a labeled training set $\mathcal{X}^L = \{\langle \mathbf{x}_1, y_1 \rangle, ..., \langle \mathbf{x}_{|\mathcal{X}^L|}, y_{|\mathcal{X}^L|} \rangle \mid \mathbf{x}_i \in \mathbb{R}^m,\ y_i \in \{-1, +1\}\}$ with a set of m-dimensional features $\mathbf{x_i}$, positive and negative labeled samples y_i and an initial uniform distribution $p(\mathbf{x_i}) = \frac{1}{|\mathcal{X}^L|}$ over the L examples. A weak classifier h is trained using $\mathcal{X}$ and $p(\mathbf{x})$. The weak classifier has to perform only slightly better than random guessing (i.e., the error rate of a classifier for a binary decision task must be less than 50%). Depending on the error e of the weak classifier, a weight $\alpha = \frac{1}{2} \ln \left(\frac{1-e}{e}\right)$ is calculated and the probability $p(\mathbf{x})$ is updated. For misclassified samples the corresponding weight is increased while for correctly classified samples the weight is decreased. Thus, the algorithm focuses on the hard examples. Boosting greedily adds a new classifier at each boosting iteration until a certain stopping criterion is met. Finally, a strong classifier $H(\mathbf{x}) = \text{sign}\left(\sum_{n=1}^{N} \alpha_n h_n(\mathbf{x})\right)$ is calculated by a linear combination of all N trained weak classifiers. As shown by Friedman et al. [21], boosting can be viewed as additive logistic regression by stage wise minimization of the exponential loss $\mathcal{L} = \sum_{\mathbf{x} \in \mathcal{X}^L} e^{-yH(\mathbf{x})}$. Thus, a confidence measure is provided by

$$P(y = 1|\mathbf{x}) = \frac{e^{H(\mathbf{x})}}{e^{H(\mathbf{x})} + e^{-H(\mathbf{x})}}. \tag{1}$$

Furthermore, boosting can be applied for feature selection [22] where each feature corresponds to a weak classifier. In each iteration n from a set of k possible features $\mathcal{F} = \{f_1, ..., f_k\}$, a weak hypothesis is built from the weighted training samples. The best f_n is selected and forms the weak hypothesis h_n. The weights of the training samples are updated with respect to the error of the chosen hypothesis.

2.2 On-Line Boosting for Feature Selection

During on-line learning, contrary to off-line methods, each training sample is only provided once to the learner and is discarded right after learning. For that purpose, the weak classifiers have to be updated on-line every time a new training sample is available. The basic idea of on-line boosting is that the importance λ of a sample can be estimated by propagating it through a fixed set of weak classifiers [23]. The importance plays the role as the weight distribution $p(\mathbf{x_i})$ in the off-line case. In fact, λ is increased proportional to the error e of the weak classifier if the sample is still misclassified and decreased, otherwise. The error of the weak classifier $\hat{e} = \frac{\lambda^w}{\lambda^w + \lambda^c}$ is estimated by the sum of correctly λ^c and incorrectly λ^w samples seen so far.

In order to perform feature selection Grabner and Bischof [8] introduced "selectors". On-line boosting is not directly performed on the weak classifiers, but on the selectors. For that purpose, a selector $h^{sel}(\mathbf{x})$ consists of a set of M weak classifiers $\{h_1(\mathbf{x}), \ldots, h_M(\mathbf{x})\}$. When training a selector, its M weak classifiers are trained and the one with the lowest estimated error is selected $h^{sel}(\mathbf{x}) = \arg\min_m e\left(h_m(\mathbf{x})\right)$. The AdaBoost on-line training algorithm used for feature selection works as follows: A fixed number of N selectors $h_1^{sel}, .., h_N^{sel}$ are initialized with random features. The weak classifiers in each selector are updated, as soon as a new training sample $\langle \mathbf{x}, y \rangle$ is available, and the weak classifier with the smallest estimated error is selected. For updating of the weak classifier any on-line learning algorithm is applicable. Finally, the weight α_n of the n-th selector h_n^{sel} is updated and the importance λ_n is passed to the next selector h_{n+1}^{sel}. Contrary to the off-line version, the on-line classifier is available at any time of the training process as a linear combination of the N selectors.

2.3 Off-Line Semi-supervised Boosting

Unsupervised methods are looking to find an interesting (natural) structure using only unlabeled data $\mathcal{X}^U = \{\mathbf{x}_1, \ldots, \mathbf{x}_{|\mathcal{X}^U|}\}$. Semi-supervised learning uses both labeled $\mathcal{X}^L$ and unlabeled $\mathcal{X}^U$ data $\mathcal{X} = \mathcal{X}^L \cup \mathcal{X}^U$. We use the recently proposed SemiBoost approach by Mallapragada *et al.* [17] which combines ideas from graph theory and clustering and outperforms other approaches on common machine learning datasets. The basic idea is to extend the loss function with unlabeled data. In order to include unlabeled samples a similarity measure $S(\mathbf{x}_i, \mathbf{x}_j)$ has to be provided to "connect" pairs of samples. The combined loss linearly combines three individual loss functions: (i) a loss for labeled examples, (ii) labeled examples and unlabeled examples and (iii) pairs of unlabeled examples. Boosting is used to minimize the combined loss.

Following the derivation of the AdaBoost algorithm the objective function is solved in a greedy manner by stage-wise selecting the best weak classifier h_n and weight α_n, which are added to the ensemble. Formally,

$$h_n = \arg\min_{h_n} \frac{1}{|\mathcal{X}^L|} \sum_{\substack{\mathbf{x} \in \mathcal{X}^L \\ h_n(\mathbf{x}) \neq y}} w_n(\mathbf{x}, y) - \frac{1}{|\mathcal{X}^U|} \sum_{\mathbf{x} \in \mathcal{X}^U} (p_n(\mathbf{x}) - q_n(\mathbf{x})) \alpha_n h_n(\mathbf{x}) \quad (2)$$

$$\alpha_n = \frac{1}{4} \ln \frac{\frac{1}{|\mathcal{X}^U|} \sum_{\substack{\mathbf{x} \in \mathcal{X}^U \\ h_n(\mathbf{x})=1}} p_n(\mathbf{x}) + \sum_{\substack{\mathbf{x} \in \mathcal{X}^U \\ h_n=-1}} q_n(\mathbf{x}) + \frac{1}{|\mathcal{X}^L|} \sum_{\substack{\mathbf{x} \in \mathcal{X}^L \\ h_n(\mathbf{x})=y}} w_n(\mathbf{x}, y)}{\frac{1}{|\mathcal{X}^U|} \sum_{\substack{\mathbf{x} \in \mathcal{X}^U \\ h_n(\mathbf{x})=1}} q_n(\mathbf{x}) + \sum_{\substack{\mathbf{x} \in \mathcal{X}^U \\ h_n(\mathbf{x})=-1}} p_n(\mathbf{x}) + \frac{1}{|\mathcal{X}^L|} \sum_{\substack{\mathbf{x} \in \mathcal{X}^L \\ h_n(\mathbf{x}) \neq y}} w_n(\mathbf{x}, y)} \quad (3)$$

where the term $w_n(\mathbf{x}, y) = e^{-2yH_{n-1}(\mathbf{x})}$, is the weight of a labeled sample. Using $\mathcal{X}^+ = \{\langle \mathbf{x}, y \rangle | \mathbf{x} \in \mathcal{X}^L, y = 1\}$ as the set of all positive samples and $\mathcal{X}^- = \{\langle \mathbf{x}, y \rangle | \mathbf{x} \in \mathcal{X}^L, y = -1\}$ as the set of all negative samples the terms

$$p_n(\mathbf{x}) = e^{-2H_{n-1}(\mathbf{x})} \frac{1}{|\mathcal{X}^L|} \sum_{\mathbf{x}_i \in \mathcal{X}^+} S(\mathbf{x}, \mathbf{x}_i) + \frac{1}{|\mathcal{X}^U|} \sum_{\mathbf{x}_i \in \mathcal{X}^U} S(\mathbf{x}, \mathbf{x}_i) e^{H_{n-1}(\mathbf{x}_i) - H_{n-1}(\mathbf{x})}, \quad (4)$$

$$q_n(\mathbf{x}) = e^{2H_{n-1}(\mathbf{x})} \frac{1}{|\mathcal{X}^L|} \sum_{\mathbf{x}_i \in \mathcal{X}^-} S(\mathbf{x}, \mathbf{x}_i) + \frac{1}{|\mathcal{X}^U|} \sum_{\mathbf{x}_i \in \mathcal{X}^U} S(\mathbf{x}, \mathbf{x}_i) e^{H_{n-1}(\mathbf{x}) - H_{n-1}(\mathbf{x}_i)} \quad (5)$$

can be interpreted as confidences of an unlabeled sample belonging to the positive ($p_n(\mathbf{x})$) and negative class ($q_n(\mathbf{x})$), respectively. The classifier is trained in order to minimize the weighted error of the samples. For a labeled sample $\mathbf{x} \in \mathcal{X}^L$ this is the same as in common boosting the weight $w_n(\mathbf{x})$. The second term considers the distance between the unlabeled sample and the labeled samples. Each unlabeled sample $\mathbf{x} \in \mathcal{X}^U$ gets the (pseudo)-label $z_n(\mathbf{x}) = \text{sign}(p_n(\mathbf{x}) - q_n(\mathbf{x}))$ and should be sampled according to the confidence weight $|p_n(\mathbf{x}) - q_n(\mathbf{x})|$.

Summarizing, the algorithm minimizes an objective function which takes distances among semi-labeled data into account using a given similarity measure between samples. When no unlabeled data is used (*i.e.*, $\mathcal{X}^U = \{\}$) Eq. 2 and Eq. 3 reduce to the well known AdaBoost formulas. After the training, we have a strong classifier similar to standard boosting.

3 Semi-supervised On-Line Boosting for Feature Selection

3.1 Approximations of the Weights

Since sample weights code the information from one weak classifier to the next, we have to determine all these weights in an on-line setting, in order to train the n-th weak classifier h_n (Eq. 2) and calculate its associated factor α_n (Eq. 3). For labeled examples we can use the on-line boosting for feature selection approach directly. The main question is how to include the unlabeled samples. Their weights and, additionally, their labels are related to $p(\mathbf{x})$ and $q(\mathbf{x})$ defined in Eq. 4 and Eq. 5, respectively. But, these terms cannot be evaluated directly, due to the sums over pairs of either labeled and unlabeled samples. Since we are in a pure on-line setting we cannot access the whole training set. Hence, we have to use approximations.

Let us assume we have a huge ($|\mathcal{X}^U| \to \infty$) amount of unlabeled data, then the second terms in Eq. 4 and Eq. 5 will be zero. Therefore, we can skip these terms without a major loss in performance. Now, following [24,18] we learn the similarity $S(\mathbf{x}_i, \mathbf{x}_j) \approx H^{sim}(\mathbf{x}_i, \mathbf{x}_j)$ by a classifier using boosting. Furthermore, we only have to sum over the similarity of the current (unlabeled) sample $\mathbf{x}$ and the set of positive or negative samples. Given the labeled samples in advance, we can train a classifier a-priori which measures the similarity, to the positive or negative class. For a positive sample this can be approximated by learning a classifier which describes the positive class $\sum_{\mathbf{x}_i \in \mathcal{X}^+} H^{sim}(\mathbf{x}, \mathbf{x}_i) \approx H^+(\mathbf{x})$, *i.e.*, provides a probability measure that $\mathbf{x}$ corresponds to the positive class. In the same manner, a classifier is built for the negative class $\sum_{\mathbf{x}_i \in \mathcal{X}^-} H^{sim}(\mathbf{x}, \mathbf{x}_i) \approx H^-(\mathbf{x})$. Instead of learning two generative classifiers we learn one discriminative classifier $H^P(\mathbf{x})$ which distinguishes the two classes, *i.e.*, $H^+(\mathbf{x}) \sim H^P(\mathbf{x})$ and $H^-(\mathbf{x}) \sim 1 - H^P(\mathbf{x})$. Since we use boosting to learn such a prior classifier, it can be translated into a probability using Eq. 1. We can now approximate Eq. 4 and Eq. 5 as

$$\tilde{p}_n(\mathbf{x}) \approx e^{-H_{n-1}(\mathbf{x})} \sum_{\mathbf{x}_i \in \mathcal{X}^+} S(\mathbf{x}, \mathbf{x}_i) \approx e^{-H_{n-1}(\mathbf{x})} H^+(\mathbf{x}) \approx \frac{e^{-H_{n-1}(\mathbf{x})} e^{H^P(\mathbf{x})}}{e^{H^P(\mathbf{x})} + e^{-H^P(\mathbf{x})}}, \quad (6)$$

$$\tilde{q}_n(\mathbf{x}) \approx e^{H_{n-1}(\mathbf{x})} \sum_{\mathbf{x}_i \in \mathcal{X}^-} S(\mathbf{x}, \mathbf{x}_i) \approx e^{H_{n-1}(\mathbf{x})} H^-(\mathbf{x}) \approx \frac{e^{H_{n-1}(\mathbf{x})} e^{-H^P(\mathbf{x})}}{e^{H^P(\mathbf{x})} + e^{-H^P(\mathbf{x})}}, \quad (7)$$

where we discard the factor 2 since we do not include pairs of unlabeled to unlabeled samples. Since we are interested in the difference, we finally get the "pseudo-soft-label"

$$\tilde{z}_n(\mathbf{x}) = \tilde{p}_n(\mathbf{x}) - \tilde{q}_n(\mathbf{x}) = \frac{\sinh(H^P(\mathbf{x}) - H_{n-1})}{\cosh(H^P(\mathbf{x}))} = \tanh(H^P(\mathbf{x})) - \tanh(H_{n-1}(\mathbf{x})). \quad (8)$$

Algorithm 1. On-line Semi-supervised Boosting for feature selection

Require: training (labeled or unlabeled) example $\langle \mathbf{x}, y \rangle$, $x \in \mathcal{X}$
Require: prior classifier H^P (can be initialized by training on $\mathcal{X}^L$)
Require: strong classifier H (initialized randomly)
Require: weights $\lambda^c_{n,m}$, $\lambda^w_{n,m}$ (initialized with 1)
for $n = 1, 2, .., N$ **do** // for all selectors
  if $x \in \mathcal{X}^L$ **then** //set weight and target of the sample
    $y_n = y$, $\lambda_n = \exp(-yH_{n-1}(\mathbf{x})$
  else
    $y_n = \text{sign}(p(\mathbf{x}) - q(\mathbf{x}))$, $\lambda_n = |p(\mathbf{x}) - q(\mathbf{x})|$ //set pseudo label
  end if
  for $m = 1, 2, .., M$ **do** // update the selector h^{sel}_n
    $h_{n,m} = \text{update}(h_{n,m}, \langle \mathbf{x}, y \rangle, \lambda)$ // update each weak classifier
    // estimate errors
    if $h^{weak}_{n,m}(\mathbf{x}) = y$ **then**
      $\lambda^c_{n,m} = \lambda^c_{n,m} + \lambda_n$
    else
      $\lambda^w_{n,m} = \lambda^w_{n,m} + \lambda_n$
    end if
    $e_{n,m} = \frac{\lambda^w_{n,m}}{\lambda^c_{n,m} + \lambda^w_{n,m}}$
  end for
  // choose weak classifier with the lowest error
  $m^+ = \arg\min_m(e_{n,m})$, $e_n = e_{n,m^+}$, $h^{sel}_n = h_{n,m^+}$
  if $e_n = 0$ or $e_n > \frac{1}{2}$ **then**
    exit
  end if
  $\alpha_n = \frac{1}{2} \cdot \ln \frac{1-e_n}{e_n}$ // calculate voting weight
end for

3.2 The On-Line Algorithm

It is now straight forward to extend SemiBoost to on-line boosting [8]. For the labeled examples $\langle \mathbf{x}, y \rangle, x \in X^L$ ($y \in \{-1, +1\}$) nothing changes. For each unlabeled sample ($x \in \mathcal{X}^U$) after each selector not only the weight (the importance λ_n) is adapted, but also its estimated target y_n may change. Hence, for unlabeled samples in each selector n, we set

$$y_n = \text{sign}(\tilde{z}_n(\mathbf{x})) \text{ and } \lambda_n = |\tilde{z}_n(\mathbf{x})|, \quad (9)$$

where $\tilde{z}_n(\mathbf{x})$ is defined in Eq. 8. Summarizing, by training a prior classifier H^P from labeled samples a-priori provided, it is possible to include unlabeled data into the on-line boosting framework using pseudo-labels and pseudo-importances. Our semi-supervised boosting algorithm for feature selection is sketched in Algorithm 1. Compared to the original on-line boosting algorithm [8] only a few lines of code (highlighted lines 2-6) have to be changed in order to cope with unlabeled data.

Let us take a look at the pseudo-labels when propagating the unlabeled sample $\mathbf{x}$ through the selectors. If the prior is very confident it dictates the label. A label switch can happen, *i.e.*, $H(\mathbf{x})$ can overrule $H^P(\mathbf{x})$, if $\tilde{z}_n(\mathbf{x})$ has a differnt label as the prior $H^P(\mathbf{x})$. As can be easily seen from Eq. 8, this is the case if $|H_n| > |H^P|$. Therefore, the more confident the prior is, the longer (with respect to n) the label is not allowed to change. We do not make any statements whether this is a correct or incorrect label switch. Note, that the prior classifier can be wrong, but it has to provide a "honest" decision. Meaning, if it is highly confident it must be ensured to be a correct decision[2].

4 Robust Tracking

We first briefly review the on-line boosting tracker that is based on on-line boosting for feature selection [8,4], which is replaced by our proposed on-line SemiBoost algorithm.

The basic idea is to formulate tracking as binary classification problem between the foreground object, which has to be tracked, and the local background. Assuming the object has been detected in the first frame, an initial classifier is built by taking positive samples from the object and randomly chosen negative ones from the background. The tracking loop consists of the following steps. From time t to $t+1$ the classifier is evaluated exhaustively pixel by pixel in the local neighborhood. Since the classifier delivers a response which is equivalent to the log-likelihood ratio $H(\mathbf{x}) = \frac{1}{2}\log\left(\frac{P(y=1|\mathbf{x})}{P(y=-1|\mathbf{x})}\right)$ (see Eq. 1), the confidence distribution is analyzed and in the simplest case the local maximum is considered to be the new object position. In order to robustly find the object in the next frame and thus adapt to appearance changes of the object, different lightning conditions or background changes, the classifier gets updated. A positive update is taken for the patch where the object is most likely to be and negative updates are drawn from the local neighborhood.

4.1 Modifications of the Tracking Loop

The tracker, as reviewed above, can suffer from the drifting problem. This is due to self-learning relies on its own predictions that are always incorporated with hard labels (*i.e.*, $y \in \{-1, +1\}$), even if their confidences are very low. In

[2] There are also relations to the co-training [25] assumptions, *i.e.*, a classifier should be never "confident but wrong".

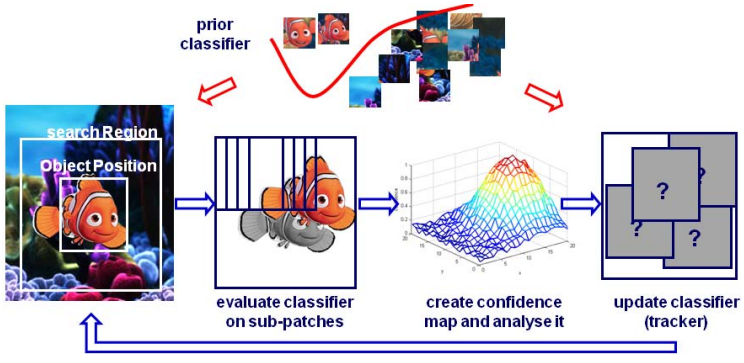


Fig. 3. Given a fixed prior and an initial position of the object in time t, the classifier is evaluated at many possible positions in a surrounding search region in frame $t+1$. The obtained confidence map is analyzed in order to estimate the most probable position and finally the tracker (classifier) is updated in an unsupervised manner, using randomly selected patches.

contrast, incorporating our novel way of on-line semi-supervised boosting allows us to change the update strategy of the previously proposed on-line boosting tracker. The overall work flow is depicted in Figure 3, which is very similar to the one described above. The main difference is, that we do not update the classifier with fixed labels, we solely use (random) patches from the region of the estimated object position and use them as unlabeled samples to update the classifier. This is only possible because we have a prior classifier. Roughly speaking, one can think of the prior classifier as a fixed point and the on-line classifier exploring the space around it. This means that the classifier can adapt (or "drift") to new situations but has always the possibility to recover.

5 Experiments and Discussion

In this section, first, we perform experiments demonstrating the specific properties of our tracking approach. Second, we evaluated our tracker on different scenarios showing that we can cope with a large variability of different objects.

As image features which are selected by on-line SemiBoost we use Haar-like features [14] which can be calculated efficiently using integral data-structures. The performance (speed) depends on the size of the search region which we have defined by enlarging the target region by one third of the object size in each direction (for this region the integral representation is computed). In our experiments we neither use a motion model nor a scaled search window, which both however can be incorporated quite easily. The strong classifier consists of only 25 selectors each with a feature pool of 50 weak classifier. All experiments are performed on a common 2.0 GHz PC with 2 GB RAM, where we achieve 25 fps tacking speed.

5.1 Illustrations

We illustrate details of our tracker on frontal faces. As prior classifier and for initialization of the tracking process we take the default frontal face detector from OpenCV Version 1.0[3]. This demonstrates that we can use any prior in our method. The primary focus of the experiments is to compare the SemiBoost tracker with other combination methods for the prior and the on-line method[4]. As can be seen from Fig. 4, our approach (second row) significantly outperforms the on-line booster, the prior classifier and a heuristic combination of prior and on-line booster (first row). Additionally, even if the prior has very low confidence (third row), the tracker is still able to correctly follow the (side) face. This shows that we can adapt to appearance changes.

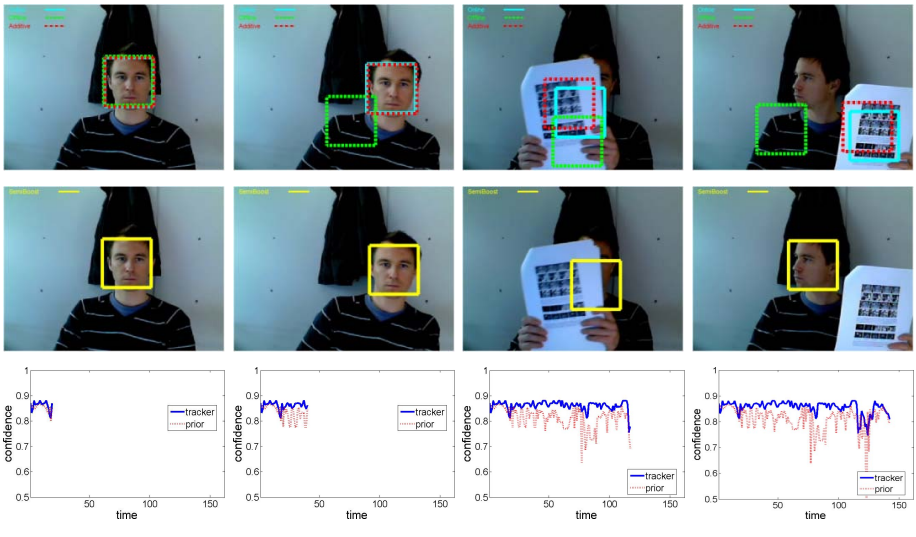


Fig. 4. Tracking a face in an image sequence under various appearance changes. The first row illustrates three different types of update strategies for the tracker, *i.e.*, (i) on-line boosting (cyan), (ii) prior classifier (red) and (iii) a heuristic combination of (i) and (ii) using the sum-rule, *i.e.*, $0.5(H^P(\mathbf{x}) + H(\mathbf{x}))$ (green). The second row shows the SemiBoost tracker using the same off-line prior. The last row depicts confidence values of the tracked patch over time for the prior and the SemiBoost tracker, respectively.

Fig. 5 depicts some illustrative samples taken for updates for both the on-line tracker and our tracker. As can be observed, while both approaches track the same object, they incorporate totally different updates. After some time the on-line booster performs wrong updates with still high confidence. This is the main reason for drifting. Furthermore, in the SemiBoost method both sample labels

[3] `http://sourceforge.net/projects/opencvlibrary/`, 2008/03/16.
[4] The OpenCV detector fails on side looking faces.

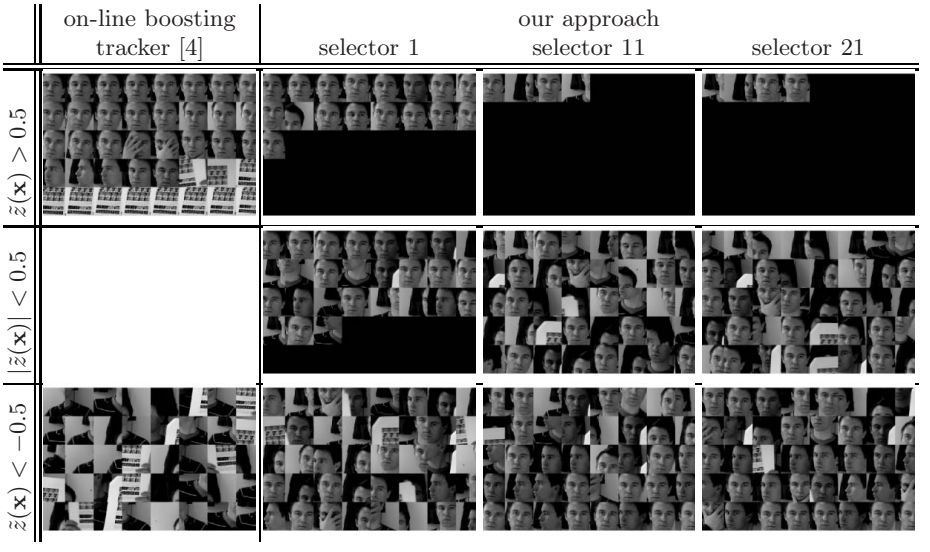


Fig. 5. Typical updates used for the former on-line boosting tracker (first column). If the tracker loses the object and due to the self-learning update strategy which delivers hard updates, it focuses on another image region. The remaining columns show how samples are incorporated by the SemiBoost tracker. While they are propagated through the selectors, their importance and label can change, with respect to the prior.

and sample weights change while propagating through the selectors of the SemiBoost tracker while being constant for the former approach. Only such samples are incorporated which are necessary to augment the prior knowledge or invert it in order to be adaptive. Positive samples are inherently treated with caution, *i.e.*, only few positive examples are considered.

5.2 One-Shot Prior Learning

For these experiments, the prior is learned from the first frame only. In fact, we build a trainingset $\mathcal{X}_P = \langle \mathbf{x}_o, +1 \rangle \cup \{ \langle \mathbf{x}_i, -1 \rangle | \mathbf{x}_i \neq \mathbf{x}_o \}$ where $\mathbf{x}_o$ corresponds to the marked image region and negative samples are generated from the local neighborhood[5]. Since this trainingset is quite small the time needed to train the prior classifier H^P is negligible. After this one-shot training, the prior classifier is kept constant.

In Fig. 6, we compare our new method to the on-line boosting approach on various tracking scenarios. First, as can be seen in row 1, we are still able to handle challenging appearance changes of the object. Row 2 of Fig. 6 depicts tracking during a fast movement. Since some incorrect updates and the self-learning strategy

[5] Also some invariance can be included in the training set, *e.g.*, by adding "virtual" samples [26] in order to train a more robust classifier.

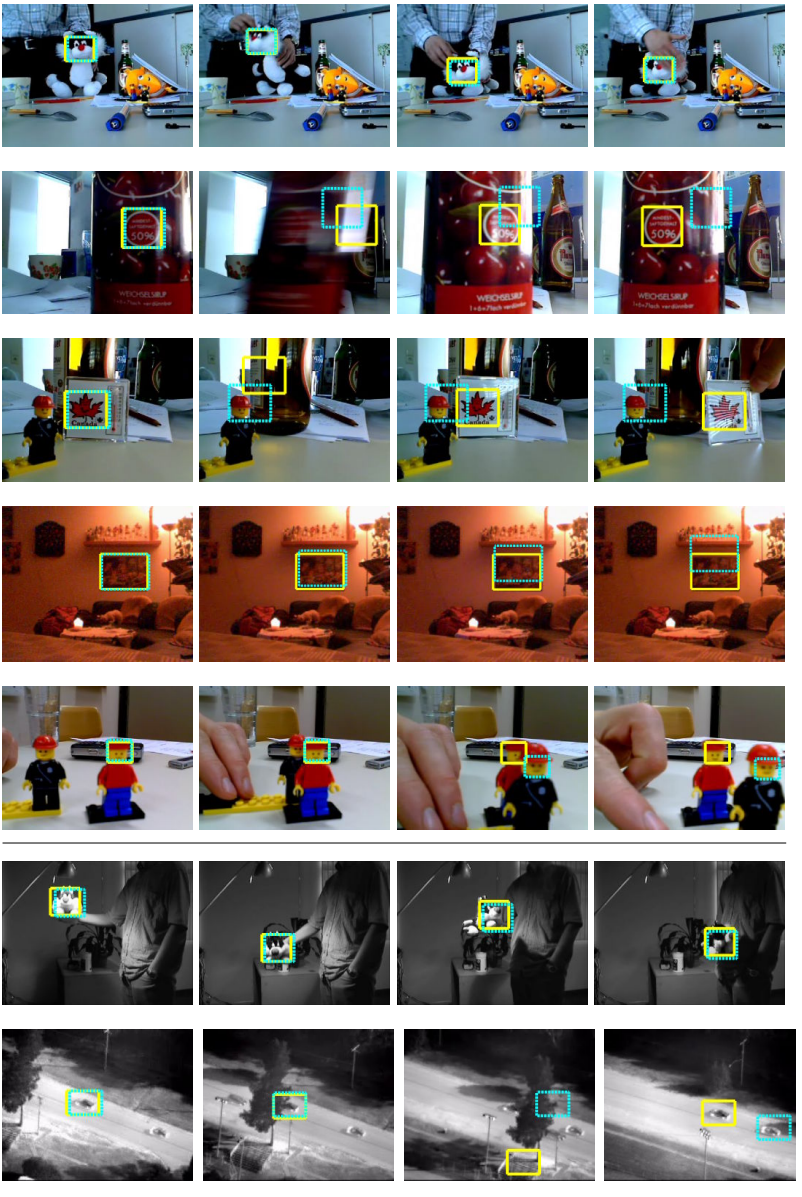


Fig. 6. Comparisons of our proposed SemiBoost tracker (yellow) and the previously proposed on-line tracker (dotted cyan). Our approach is still able to adapt to appearance changes while limiting the drinfing. Additionally, results on two public sequences are shown (last two rows). The first sequence have been provided by Lim and Ross ([6]) and the second sequence is taken from the VIVID-PETS 2005 dataset[6].

[6] `http://www.vividevaluation.ri.cmu.edu/datasets/datasets.html`, 2007/06/12

of the on-line boosting tracker loses the target and focuses on another part while the semi-supervised tracker is able to re-detect the object. An extremal case is shown in row 3, where we remove the object from the scene. If the object is present again and thanks to the fixed prior our proposed approach has not forgotten the appearance as it is the case for the other tracker and snap to the object again. The next experiment (row 4) focuses on the long term behavior. We chose to track a non-moving object in a static scene for about 1 hour. In order to emphasize the effect we use rather dark illumination conditions. While our proposed tracker stays at the object, the on-line booster starts to drift away. The reason is the accumulation of errors. The final experiment shows a special case of drifting as depicted in the last row of Fig. 6. Two very similar objects are put together in the scene. Since the pure on-line tracker has not the additional prior information, it is very likely that it is unstable and may switch to another object. Additionally, we choose two public available tracking sequences which have been already used in other publications as can be seen in the last two rows. Our approach performs comparable to the previous on-line tracker on appearance changes (sixth row). After the object was totally occluded (last row), our approach is able to recover the correct object while the former on-line tracker gets confused and starts tracking the second (wrong) car. Additional tracking videos are included as supplementary material.

6 Conclusions

In this paper, we have presented a tracker which limits the drifting problem while still being adaptive to various appearance changes which arise in typical real world scenarios. We have employed ideas from semi-supervised learning and on-line boosting for feature selection. The so trained on-line classifier is used in a tracking framework in order to discriminate the object from the background. The knowledge from labeled data can be used to build a fixed prior for the on-line classifier. In order to still be adaptive, during tracking unlabeled data is explored in a principled manner. Furthermore, our approach does not need parameter tuning and is easy to implement. We have demonstrated successful tracking of different objects in real-time on various challenging sequences.

References

1. Avidan, S.: Support vector tracking. IEEE Trans. PAMI 26, 1064–1072 (2004)
2. Lepetit, V., Lagger, P., Fua, P.: Randomized trees for real-time keypoint recognition. In: Proc. CVPR, vol. 2, pp. 775–781 (2005)
3. Özuysal, M., Fua, P., Lepetit, V.: Fast keypoint recognition in ten lines of code. In: CVPR (2007)
4. Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via on-line boosting. In: Proc. BMVC, vol. 1, pp. 47–56 (2006)
5. Collins, R., Liu, Y., Leordeanu, M.: Online selection of discriminative tracking features. IEEE Trans. PAMI 27(10), 1631–1643 (2005)

6. Lim, J., Ross, D., Lin, R., Yang, M.: Incremental learning for visual tracking. In: Saul, L.K., Weiss, Y., Bottou, L. (eds.) NIPS, vol. 17, pp. 793–800. MIT Press, Cambridge (2005)
7. Avidan, S.: Ensemble tracking. In: Proc. CVPR, vol. 2, pp. 494–501 (2005)
8. Grabner, H., Bischof, H.: On-line boosting and vision. In: Proc. CVPR, vol. 1, pp. 260–267 (2006)
9. Matthews, I., Ishikawa, T., Baker, S.: The template update problem. IEEE Trans. PAMI 26, 810–815 (2004)
10. Grabner, M., Grabner, H., Bischof, H.: Learning features for tracking. In: Proc. CVPR (2007)
11. Tang, F., Brennan, S., Zhao, Q., Tao, H.: Co-tracking using semi-supervised support vector machines. In: Proc. ICCV, pp. 1–8 (2007)
12. Woodley, T., Stenger, B., Cipolla, R.: Tracking using online feature selection and a local generative model. In: Proc. BMVC (2007)
13. Grossberg, S.: Competitive learning: From interactive activation to adaptive resonance. Neural networks and natural intelligence, 213–250 (1998)
14. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. CVPR, vol. I, pp. 511–518 (2001)
15. Li, Y., Ai, H., Yamashita, T., Lao, S., Kawade, M.: Tracking in low frame rate video: A cascade particle filter with discriminative observers of different lifespans. In: Proc. CVPR, pp. 1–8 (2007)
16. Zhu, X.: Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison (2005)
17. Mallapragada, P.K., Jin, R., Jain, A.K., Liu, Y.: Semiboost: Boosting for semi-supervised learning. Technical report, Department of Computer Science and Engineering, Michigan State University (2007)
18. Leistner, C., Grabner, H., Bischof, H.: Semi-supervised boosting using visual similarity learning. In: Proc. CVPR (to appear, 2008)
19. Schapire, R.: The boosting approach to machine learning: An overview. In: Proceedings MSRI Workshop on Nonlinear Estimation and Classification (2001)
20. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1), 119–139 (1997)
21. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. Annals of Statistics 28(2), 337–407 (2000)
22. Tieu, K., Viola, P.: Boosting image retrieval. In: Proc. CVPR, pp. 228–235 (2000)
23. Oza, N., Russell, S.: Online bagging and boosting. In: Proceedings Artificial Intelligence and Statistics, pp. 105–112 (2001)
24. Hertz, T., Bar-Hillel, A., Weinshall, D.: Learning distance functions for image retrieval. In: Proc. CVPR, vol. 2, pp. 570–577 (2004)
25. Balcan, M.F., Blum, A., Yang, K.: Co-training and expansion: Towards bridging theory and practice. In: NIPS. MIT Press, Cambridge (2004)
26. Girosi, F., Chan, N.: Prior knowledge and the creation of virtual examples for rbf networks. In: IEEE Workshop on Neural Networks for Signal Processing (1995)

Reformulating and Optimizing the Mumford-Shah Functional on a Graph — A Faster, Lower Energy Solution

Leo Grady and Christopher Alvino

Siemens Corporate Research — Department of Imaging and Visualization
755 College Road East, Princeton, NJ 08540

Abstract. Active contour formulations predominate current minimization of the Mumford-Shah functional (MSF) for image segmentation and filtering. Unfortunately, these formulations necessitate optimization of the contour by evolving via gradient descent, which is known for its sensitivity to initialization and the tendency to produce undesirable local minima. In order to reduce these problems, we reformulate the corresponding MSF on an arbitrary graph and apply combinatorial optimization to produce a fast, low-energy solution. The solution provided by this graph formulation is compared with the solution computed via traditional narrow-band level set methods. This comparison demonstrates that our graph formulation and optimization produces lower energy solutions than gradient descent based contour evolution methods in significantly less time. Finally, by avoiding evolution of the contour via gradient descent, we demonstrate that our optimization of the MSF is capable of evolving the contour with non-local movement.

1 Introduction

The Mumford-Shah functional (MSF) formulates the problem of finding piecewise smooth reconstructions of functions (e.g., images) as an optimization problem [1]. Optimizing the MSF involves determining both a function and a contour across which function smoothness is not penalized. Unfortunately, since smoothness of the reconstruction is not enforced across the contour and since the contour is variable in the optimization, the functional is not easily minimized using classical calculus of variations.

Given a fixed contour it is possible to solve for the optimal reconstruction function by solving a straightforward elliptic PDE with Neumann boundary conditions. Additionally, given a fixed piecewise smooth reconstruction function, it is possible to determine, at each point on the contour, the direction that the contour would move to decrease the functional as quickly as possible. Thus, most methods for solving the MSF involve alternating optimization of the reconstruction function and the contour [2,3,4]. The results of performing this type of optimization are well known and achieve satisfactory results that are used for different imaging applications [4]. Unfortunately, this optimization of the MSF using contour evolution techniques (typically implemented with level sets) is slow primarily due to the small steps taken by the contour at each iteration. This slowness is exacerbated by the fact that a small perturbation of the contour can have a relatively large effect on the optimal reconstruction function. Additionally, these

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 248–261, 2008.

traditional methods often require many implementation choices (e.g., implementation parameters) and these choices may produce differences in the final result.

Practical energy minimization problems formulated on a finite set of variables can be solved efficiently using combinatorial algorithms [5,6,7]. Furthermore, because of the well-established equivalence between the standard operators of multidimensional calculus and certain combinatorial operators, it is possible to rewrite many PDEs formulated in ^{N} equivalently on a complex (graph). Reformulating the conventional, continuous, PDE on a graph permits straightforward application of the arsenal of combinatorial optimization techniques to efficiently solve these variational problems. An alternate view of our approach is to consider rewriting the continuous energy functional in terms of the precise discrete operations that would be performed on a computer to evaluate the energy of a particular solution. By writing this energy in discrete terms, we can design our optimization method to optimize the energy value that would actually be measured by the computer. In this work, we reformulate the difficult MSF on a graph so that we may apply a combinatorial optimization to reduce the difficulties of speed and local minima associated with the small contour improvements obtained via traditional contour evolution. An added benefit of reformulating an energy in a combinatorial setting is that such a generic formulation may be applied without modification to higher dimensional data or general data analysis problems, such as point clustering, mesh smoothing/segmentation or space-variant vision.

Graph based optimization techniques have previously been used as components in optimization methods for functionals formulated in continuous space. Boykov *et al.* suggest using a max-flow step to assist in level set updates [8]. Zeng *et al.* [9] and El-Zehiry *et al.* [10] employ a max-flow operation as a component of their minimization of the piecewise constant MSF; we instead present a complete combinatorial reformulation and solution of the more general piecewise smooth MSF. Likewise, graph techniques have also been employed in the minimization of total variation methods [11].

Traditional contour evolution optimizations pursue a contour update in the direction of the highest gradient. Since this contour update represents a first variation of the MSF, calculation of this update does not require knowledge of the idealized foreground and background functions (images) at locations distant from the contour. In contrast, our graph formulation leads us to a combinatorial optimization approach that is capable of taking arbitrarily large steps of the contour location. Taking these large steps requires us to address the estimation of the foreground/background function values at locations (pixels) distant from the contour. To the knowledge of the authors, this work represents the first proposal of extending these foreground and background functions outside their region of evaluation.

2 Method

In this section, we define the continuous piecewise smooth Mumford-Shah model that we use, and go through each energy term to formulate the combinatorial analogue of the piecewise smooth MSF. With these combinatorial analogues, we then proceed to show how to optimize the foreground/background reconstruction and contour location.

2.1 Mumford-Shah Formulation: Continuous and Combinatorial

A **graph** consists of a pair $G = (V, E)$ with **vertices (nodes)** $v \in V$ and **edges** $e \in E \subseteq V \times V$, with $N = |V|$ and $M = |E|$. An edge, e, spanning two vertices, v_i and v_j, is denoted by e_{ij}. A **weighted graph** assigns a value to each edge called a **weight**. The weight of an edge, e_{ij}, is denoted by $w(e_{ij})$ or w_{ij} and is assumed to be nonnegative. The **degree** of a vertex is $d_i = \sum w(e_{ij})$ for all edges e_{ij} incident on v_i. The following will also assume that our graph is connected and undirected (i.e., $w_{ij} = w_{ji}$). An image may be associated with a graph by identifying each pixel with a node and defining an edge set to represent the local neighborhood of the pixels (e.g., a 4-connected lattice).

Since the inception of the Mumford-Shah functional, there have been several related notions of what constitutes *the* Mumford-Shah functional. In this work, we follow the level set literature to consider the piecewise smooth model [1,4], formulated as

$$E(f, g, R) = \alpha \left(\int_R (f - p)^2 + \int_{\Omega \setminus R} (g - p)^2 \right) + \mu \left(\int_R ||\nabla f||^2 + \int_{\Omega \setminus R} ||\nabla g||^2 \right) + \nu \Gamma(R), \quad (1)$$

where Ω represents the image domain, f is the smoothed foreground function, g is the smoothed background function, R is the region of the image comprising the foreground, p is the pixel intensity, $\Gamma(R)$ is a function returning the length of the contour of region R, and α, μ, ν are free parameters. We assume that the image consists of grayscale values only, although the formulation can be extended to color or multispectral images. To simplify the parameter space, we assume that all three free parameters are strictly positive and divided by the value of μ. Thus, we will omit the inclusion of μ in the remaining exposition. Similar models were considered by Blake and Zisserman, who referred to the energy as the "weak membrane model" [12] and by the influential paper of Geman and Geman [13].

Formulation of (1) on a graph requires the use of combinatorial analogues of the continuous vector calculus operators (for an introduction to these combinatorial analogues, see [14]). Although combinatorial representations of differential operations are fairly well established, the challenge in the graph reformulation of any particular energy (or PDE) is to associate variables in the continuous formulation with representative combinatorial structures (pixels, edges, cycles, etc.) and, as in the continuous case, to produce a useful representation of a "contour". Specifically, each integral may be considered as a pairing between a chain (domain of integration) and cochain (function to be integrated). Associating each pixel in our image with a node in the graph, the integration over a collection of pixels (in set $S_R \subseteq V$) may be represented by the $N \times 1$ chain vector r, where

$$r_i = \begin{cases} 1 & \text{if } v_i \in S_R, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

The other two variables in E are cochains taking real values, i.e., $f_i \in \quad , g_i \in \quad$. Note also that the image I is treated as a vectorized, real-valued cochain existing on the nodes (pixels). Both chains and cochains will be treated as column vectors.

The first (data) term in (1) concerns quantities associated with pixels (i.e., intensities). We chose above to associate nodes with pixels, so p, f, and g must represent a 0-cochain (a function mapping nodes to real numbers). This matches the continuous conception of these quantities as scalar fields. Since the data term in (1) integrates over a set of the domain for which p, f and g are defined, r must represent a 0-chain indicating a region of the domain. Thus, the analogue of the first term on a graph is

$$E_1(f,g,r) = r^T (f-p)^2 + (1-r)^T (g-p)^2 \ . \tag{3}$$

In order to formulate the second term, recall that the combinatorial analogue of the gradient operator is given by the node-edge incidence matrix, A, [14]. Consequently, we may write the gradient of f as the product Af. However, since gradients are *vector functions* (corresponding to cochains on edges in the combinatorial setting) and the integral in the second term is performed over a *scalar* function (i.e., the norm of the gradient at each point), we have to transfer the gradient cochain associated with edges back to a scalar cochain associated with nodes. Such an operator may be represented by the absolute value of the incidence matrix, although each edge is now double counted, requiring a normalizing factor of one-half. Specifically, the second term may be formulated as

$$E_2(f,g,r) = \frac{1}{2}\left(r^T |A|^T (Af)^2 + (1-r)^T |A|^T (Ag)^2\right). \tag{4}$$

Finally, the contour length term may be formulated on a graph by counting the edges spanning from R to $\overline{R}$. Such a measure may be represented in matrix form as

$$E_3(f,g,r) = 1^T |Ar|. \tag{5}$$

If our graph is a standard 4-connected lattice, then (5) produces the ℓ_1 measure of the contour of region R. If we view the graph as embedded in ^{N} and wish to measure Euclidean contour length, it was shown [15] that a suitably weighted graph and corresponding incidence matrix could instead be used in (5). However, since this construction was designed to produce a Euclidean contour length, we use this construction only in term E_3. For purposes of generality and clarity here, we will continue to use the same A in all terms.

All three terms may now be put back together to define the combinatorial analogue of the piecewise smooth Mumford-Shah model, i.e.,

$$E(f,g,R) = \alpha\left(r^T (f-p)^2 + (1-r)^T (g-p)^2\right) + \frac{1}{2}\left(r^T |A|^T (Af)^2 + (1-r)^T |A|^T (Ag)^2\right) + \nu 1^T |Ar|. \tag{6}$$

Given the above definition of the combinatorial analogue of the Mumford-Shah functional, we now proceed to show how to optimize the variables f, g and r.

2.2 Optimization

We adopt the alternating optimization procedure common to optimization of the MSF [3,4]. The alternating optimization procedure first treats the current contour, r, as fixed

and then finds the optimal f, g. Given an f and g, the optimal r may then be found. We begin by considering the production of an optimal f and g from a fixed contour, r.

Taking a partial derivative of (6) with respect to f yields

$$\frac{\partial E}{\partial f} = 2\alpha \operatorname{diag}(r)\,(f - p) + A^T \operatorname{diag}\left(|A|r\right) Af. \tag{7}$$

The $\operatorname{diag}(\cdot)$ operator represents the diagonal matrix formed by placing the argument vector on the diagonal. Since both the first and second terms of (6) are positive semi-definite, the zero of (7) represents a minimum of (6). Therefore, the optimal f given a contour satisfies

$$\left(2\alpha \operatorname{diag}(r) + A^T \operatorname{diag}\left(|A|r\right) A\right) f = 2\alpha \operatorname{diag}(r) p. \tag{8}$$

We pause to consider the interpretation of this optimum for f. The construction $A^T C A$, for diagonal C, produces the Laplacian matrix with edge weights given by the diagonal of C [14]. Note that in the standard conception of the MSF, these weights all equal unity — the domain is homogeneous, except at the contour. Consequently, (8) can be interpreted within region R as solving for the f that would be produced from initializing f within R to the image and then running linear isotropic diffusion for time equal to $\frac{1}{\alpha}$.

Outside of region R, any values of f will satisfy (8). In the computation of the energy in (6) this part of f does not contribute and may be ignored. In fact, in the existing literature, the values of f outside region R are never considered, since an infinitesimal gradient step is being taken by the contour of the level set function and values of f distant from the contour are inconsequential. However, in our combinatorial formulation, we desire to take an *optimal* contour step, regardless of the proximity of the new contour to the previous contour. Consequently, we will need to produce a meaningful f outside of region R. An important assumption about f is that it is a continuous function as it approaches the contour. Therefore, in order to enforce maximum smoothness between f inside R and the extended f outside of R, we propose to construct f outside of region R to satisfy the Laplace equation while treating the f inside of R (from (8)) as Dirichlet boundary conditions. We extend g inside of R similarly. Note, however, that other extensions of f and g are possible and may lead to improved performance. Using this construction, we may produce the optimal f inside the region as

$$\left(\alpha I + L_R\right) f_R = \alpha p_R, \tag{9}$$

where I is the identity matrix and L_R indicates the portion of the Laplacian matrix corresponding to the region R. Recall that the Laplacian matrix is defined $L = A^T C A$, for some diagonal matrix C taking the edge weights along the diagonal. We may solve the Laplace equation on a general graph, given boundary conditions, by using the technique of [16], which requires the solution to a linear system of equations with a subset of the Laplacian. Using the same procedure, the optimal g_R is given by solving the system

$$\left(\alpha I + L_{\overline{R}}\right) g_{\overline{R}} = \alpha p_{\overline{R}}, \tag{10}$$

and g_R may also be found by solving the combinatorial Laplace equation as in [16].

We can now address the optimization of r, given a fixed f and g. Noting that all three terms of (6) are submodular linear functions of r, we can solve for r as a max-flow/min-cut

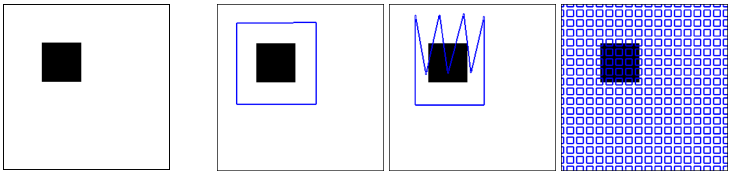
computation [6]. In effect, the first and second terms describe unary terms penalizing data infidelity from the reconstructed image and nonsmoothness in the reconstructed image. The third term penalizes contour length and is written in terms of strictly positive weights, producing a submodular energy that may be optimized effectively with a max-flow/min-cut computation. Minimum cut computations on graphs representing images are very fast using the algorithm of Boykov and Kolmogorov [17].

We conclude the section with observations about our graph formulation compared with discretized contour evolution approaches of the continuous energy. First, in contrast with standard continuous methods, at each iteration we are solving for a reconstructed function and contour that *optimally* minimize the MSF given a fixed contour or a fixed reconstruction, respectively. Due to these globally optimal steps, *all correct implementations will produce an equivalent answer*. Thus, there is no need for any implementation parameters. For example, any linear system solver will produce the same answer to (9). One method may be faster than another, but both methods will produce the same answer if run to convergence; thus, there is no need to be concerned that implementation choices will affect the *quality* of the solution. Second, because our contour optimization is not performed via gradient descent, the contour can move non-locally, and "snap" to the lowest energy contour, even for distant initializations. This non-local movement results in greater robustness to initialization, far fewer iterations and greater robustness to choice of the three term weightings in the MSF. Additionally, as shown in Section 3.3, this non-local movement capability allows our formulation to jump over intervening structures of arbitrary size to find a low-energy solution to the MSF.

2.3 Relationship to Graph Cuts

The Graph Cuts algorithm for image segmentation/denoising was first introduced in [18,19]. This algorithm has been greatly extended since inception to where it is somewhat unclear what comprises "Graph Cuts". However, all algorithms under the title "Graph Cuts" seem to have the following qualities: 1) Defined on a (possibly directed) graph, 2) Using submodular edge weights to reflect likely contour locations, 3) Possibly including an intensity prior assigning each pixel to foreground/background, 4) Possibly including hard constraints (seeds) to force pixels to be foreground or background, 5) Optimization via a max-flow/min-cut computation, 6) Produces a global optimum of the desired energy.

With this definition of "Graph Cuts", we observe that the contour optimization in the combinatorial formulation of the MSF, (6), shares much in common. Specifically, intensity priors are present (from the data term), the weights are submodular and the optimum of (6) is obtained via a max-flow/min-cut computation. However, by examining the above list, one may also notice differences with the combinatorial MSF. First, the edge weights are not modified to reflect image gradients. Second, in addition to the intensity priors, (6) involves an additional unary term penalizing the estimate of the normalized gradient near the pixel (from the smoothness term in the MSF). Third, no hard constraints (seeds) are imposed to constrain the labeling of any pixels. Fourth, there is no reconstructed image variable (i.e., f, g) present in Graph Cuts. Finally, the solution of r (contour) is just one part of one iteration in the overall optimization of the MSF. Although the solution of r is optimal for each iteration, the overall energy minimization



(a) Toy image (b) Box (c) Crown (d) Squares

Fig. 1. Toy image to compare the speed of traditional contour evolution implementation with proposed combinatorial optimization of the Mumford-Shah functional presented in this paper. The contours (blue) indicate different initializations used to generate the results in Table 1.

of the MSF still produces a local minimum. It should be noted that certain extensions of the Graph Cuts work also use Graph Cuts as a subroutine while re-estimating the intensity priors at each iteration [20]. However, unlike the MSF, this work does not include a specific smoothness penalty or a reconstructed image, but hard constraints are included and the edges are weighted by image gradients.

3 Results

The positives and negatives of MSF segmentation and reconstruction have been well-discussed in the literature. Our reformulation of the MSF on a graph is intended to permit the usage of combinatorial optimization methods to minimize the MSF more quickly and to find lower-energy solutions. Consequently, our experiments are dedicated to answering the following questions about the merits of traditional contour evolution optimizations of the MSF with the proposed combinatorial optimization applied to our graph formulation:

1. *Speed*: Which procedure finds a solution with fewer iterations? What is the relative cost per iteration? What is the dependence of performance on resolution?
2. *Initialization*: Which procedure is more robust to initialization of the contour?
3. *Parameters*: Which procedure is more robust to the choice of parameter settings?
4. *Energy minimization*: Which procedure produces solutions with lower energy?

To address the first three questions, we begin with a toy image of a black square on a white background. Such a trivial image was chosen since 1) There is a clear energy minimum, 2) A relatively smooth energy landscape, 3) The same answer for a wide range of parameters, 4) A clear stopping criterion (i.e., when the contour matches the square). For these reasons, we can perform controlled experiments to probe the answers to the questions posed above about the relative performance of traditional contour evolution implementations (via level sets) and our new graph formulation of the MSF.

We compared the combinatorial optimization of our graph formulation method with an efficient narrow-band level set implementation of the continuous formulation similar to the one presented in [4], although the original piecewise-smooth level set implementation was presented in [3]. Great care was taken to ensure the correctness and efficiency of the level set implementation so that a fair and accurate comparison could be made between the two methods. The method employed alternating optimizations of the contour

Table 1. Results of experiment comparing speed of convergence for level set (LS) solver and our graph (GR) formulation. Note: 1) The parameter settings were chosen to *best favor the level set method* in every experiment, 2) Exactly the same initializations were given to both algorithms, 3) The size and spacing of the squares initialization was chosen to favor the LS method. Time reported "per iteration" refers to update of the contour location, since computation of the reconstructed image is the same in both methods (although this computation is effectively doubled for GR since the inside/outside functions are extended beyond their respective region). Note that while the displayed number of level set iterations may seem particularly high, it is important to note that the initializations in these cases are very distant from the desired contour.

Initialization/Resolution	LS iterations	LS mean iter. time	GR iterations	GR mean iter. time
Box (64×64)	41	0.002s	2	0.0064s
Box (128×128)	126	0.0057s	2	0.0211s
Box (256×256)	140	0.0199s	2	0.0838s
Crown (64×64)	262	0.0023s	4	0.0091s
Crown (128×128)	1393	0.0061s	3	0.0239s
Crown (256×256)	110	0.0245s	4	0.1019s
Squares (64×64)	294	0.0072s	3	0.0094s
Squares (128×128)	940	0.0112s	3	0.0295s
Squares (256×256)	540	0.0624s	3	0.1177s

evolution and of the smooth functions as in the graph method and as has been used in all MSF minimizations of which we are aware. For efficiency, the level set function was computed and stored only in a narrow band around the contour, in which we maintained the sub-pixel position of the contour. Force extensions were computed on pixels which neighbored the contour as illustrated in [21]. When computing the level set function update, the spatial derivatives associated with the curvature term were computed with central differences, and the spatial derivatives associated with the data terms were computed with the numerical scheme detailed in [22] to ensure that the viscosity solution was obtained for the portion of the level set evolution that is a Hamilton-Jacobi equation. At each contour evolution step, we updated with an explicit forward-Euler scheme in which the maximally stable time step was taken to ensure both stability and speed of the level set function evolution.

Our implementation of max-flow/min-cut was taken directly from the online code of Vladimir Kolmogorov. In order to produce a comparable comparison between the level set optimization and our graph framework in these 2D experiments, we choose to calculate contour length of the cut with respect to a Euclidean measure in (5) by using the weighted incidence matrix of the graph corresponding to the construction of Boykov and Kolmogorov [15] with an approximation to the Euclidean distance represented by a neighborhood connected with a distance of two pixels.

3.1 Speed and Initialization

Our first experiment examines the relative speed of traditional level set implementations and our new graph formulation for the box image using various image resolutions and contour initializations. In this experiment, we created three initializations — A larger

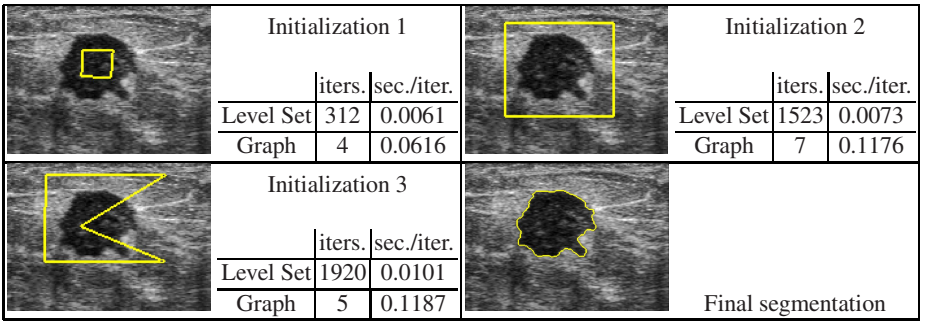


Fig. 2. Comparison of number of iterations and speed of iteration for different initializations on ultrasound image. Parameters were chosen to best benefit the level set method.

square surrounding the target square, an erratic "crown"-shaped initialization centered on the target square and small squares tiled throughout the image. These three initializations are displayed in Figure 1. For each of these initializations, we measured the number of iterations required to converge the level set and graph methods to the known optimum solution and the average time taken to produce one contour update for each method when run on an Intel Xeon 2.40GHz processor with 1GB of RAM. In this experiment, the parameters in the energy functional were chosen to favor the level set method as much as possible.

Table 1 displays the results of this experiment. The time reported "per iteration" in this table refers to the update of the contour location, since the computation of the reconstructed image is the same in both methods (although this computation is effectively doubled in our graph method since the inside/outside functions need to be extended beyond their region). Therefore, even though each iteration of our graph method is slightly more expensive than an iteration of the level set method, the improvement of 1–3 orders of magnitude in the number of iterations causes the total runtime of the graph method to be much less than that of the level set method. Additionally, the graph method converges within 2–4 iterations regardless of the resolution, initialization or parameters. Note that while the displayed number of level set iterations may seem particularly high, it is important to note that these initializations are very distant from the contour.

These experiments suggest that the proposed combinatorial optimization of the MSF produces a solution much faster than the traditional level set optimization, regardless of the resolution or contour initialization. We remind the reader that the energy term parameters were chosen to favor the level set method. Choosing the parameters to favor the proposed graph method would have resulted in an even stronger disparity in favor of our method.

A third experiment was performed on a real ultrasound image in the same manner as the first. Initializations were introduced inside the target object, outside the object and then erratically inside and outside the object. The results in terms of number of iterations and speed of each iteration are shown in Figure 2 and correspond well with the results from our synthetic experiment. Once again, the parameters of the terms in the MSF were chosen to favor the level set method and both methods converged to roughly the same contour.

3.2 Parameter Robustness

The choices of the term parameters in (1) can make drastic differences in the optimal contour and reconstruction produced by minimizing the MSF. Even if the optimal contour and reconstruction are the same for different choices of parameters, the parameter choices could affect the speed of convergence for a given initialization. In this experiment, we examine the robustness of both the contour evolution and graph formulations of the MSF to the choice of parameters in terms of the number of iterations needed to reach the optimum solution. Once again, we employ the toy example of Figure 1. For this experiment, we used the most simple, "box", initialization of Figure 1 since we expect that both algorithms will reach the target contour for all parameter choices. We ran fifty iterations in which the parameters for each of the three terms of (1) were chosen independently from a uniform distribution within the interval of zero to one and then both the level set and graph algorithms were applied to minimize the MSF. If the target square was not the optimum solution for the randomly generated parameters, this parameter set was rejected and the trial re-run. After each parameter set, the number of iterations and average time per iteration were recorded.

Table 2. Comparison of robustness to the three term parameters in (1). Using the (128×128) toy image above with the "box" initialization, for 50 trials we randomly chose the three term parameters from independent uniform distributions on the interval $(0, 1)$ and ran both the level set and graph optimizations of the MSF. A set of parameters was rejected and re-run if the target square was not the minimum of the MSF. For all parameters, the graph optimization produced the target square in 2 iterations. Note that the number of iterations reported for the level set method in Table 1 was much less than the averages reported here because all of the results reported in Table 1 used parameters that were hand-selected to favor the level set convergence.

Optimization algorithm	Mean iterations	Median iterations	Iteration number standard deviation
Level set	1614.40	1520	391.80
Graph	2	2	0

The results of this experiment are displayed in Table 2. We see that the rate of convergence of the level set method is highly dependent on the parameters, while the rate of convergence for the graph method is completely independent of the parameter set. Both algorithms exhibited independence of the per iteration time on the parameter set. Empirically, the results of this experiment concur with our experience that the convergence rate, and solution achieved, of the graph method is much less sensitive to the parameter settings than the level set method. Note that the number of iterations reported for the level set method in Table 1 was much less than the averages reported in Table 2 due to the fact that all of the results reported in Table 1 used parameters that were hand-selected to favor the level set convergence.

3.3 Non-local Movement

A key advantage of the contour optimization in our graph reformulation of the MSF is that it enables movement to the optimal location at each iteration. For this reason,

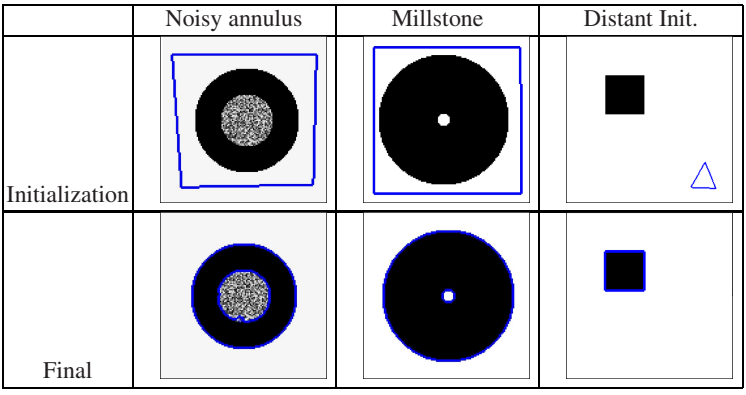


Fig. 3. Non-local movement: Since the contour optimization step in our graph formulation of the Mumford-Shah energy is not performed by gradient descent, the final contour is permitted to jump to a location distant from the initial contour location. While this effect is sometimes achieved in the level set literature by the use of mollified region indicator functions, we note that the mollifier support must be wider than the width of the annulus for such an approach to succeed.

our method is able to move to arbitrary image locations as predicted by the solution to (6) depending on the current estimate of the reconstruction functions. The motion of the contour is thus *not limited to local movements* as are traditional optimizations of the contour by gradient descent. Figure 3 illustrates three situations that are segmented correctly by the combinatorial optimization of the MSF, but where standard gradient descent methods fail.

The piecewise smooth MSF may drive non-local movement via insufficient smoothness, permitting the penetration of an annulus with a center comprised of pure noise. The final segmentation in Figure 3 is not achievable by gradient descent of the contour.

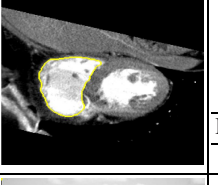
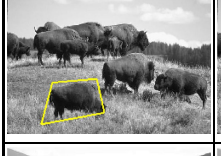
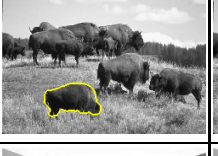
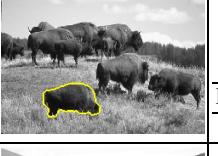
In the millstone image, we are able to achieve correct segmentation of the inner ring instantly. We would like to draw attention to the method by which Chan and Vese [2] were able to determine inner boundaries of objects. The ability to segment this inner boundary was only due to the mollified Heaviside function that was used to approximate a region indicator function. Indeed, the ability to achieve segmentation of an inner boundary in this manner is limited by the width of the resulting mollified Delta function, a quantity which should be kept low to maintain accuracy. We should also point out the work of [23] who indicate the ability to naturally attain such inner boundaries due to their method of total variation optimization of a modified MSF. Some level of non-local movement in solving the MSF with level set methods have been achieved in [24] using additive operator splitting [25], however they only illustrated the technique for the piecewise constant case.

Finally, we illustrate that distant (non-overlapping) initializations are not a problem to the combinatorial method as they are in gradient descent methods. Such a poor initialization could occur via automatic initialization of outlier image data. Regardless of the distance of the initialization from the object, our optimization is able to quickly ascertain such salient object boundaries.

3.4 Energy Minimization

Beyond speed, our purpose in introducing combinatorial optimization techniques for solving the MSF is to produce solutions with a lower energy than the solutions obtained by conventional level set techniques. In order to compare solutions in terms of minimal energy, we must address natural images for which the energy landscape is nontrivial. In this section we apply both the graph-based and level set algorithms to natural images using the same initialization/parameters and then compare the MSF energy obtained by the final solutions. The energy value is measured in exactly the same way for both algorithms — By evaluating (6). Using (6) to evaluate the energy might at first seem to be biased toward producing lower energies for the graph-based technique. However, it is important to realize that (6) details *in finite matrix form, the operations actually employed on a computer for estimating gradients and contour length*, with gradients computed by finite differences.

Our next experiment empirically compares the energy obtained for the solution of both optimizations for a variety of natural images, given the same initialization and parameters. For each image, initializations and parameters were selected to produce a contour (for at least one algorithm) that was semantically meaningful. Results of this experiment are shown in Figure 4. In every case, optimization of our graph formulation of the MSF produced solutions with an equal or lower energy and in less time.

Initialization	Level Set	Graph	Summary
			Image 1 (256×256)
			Image 2 (321×479)
			Image 3 (321×481)
			Image 4 (1488×1984)

	iters.	MS energy
Level Set	2000	40.4041
Graph	7	37.2971

	iters.	MS energy
Level Set	330	34.8106
Graph	5	34.7846

	iters.	MS energy
Level Set	1508	14.1903
Graph	6	13.7786

	iters.	MS energy
Level Set	7812	428.44
Graph	17	428.44

Fig. 4. Comparison of the energy obtained for a solution produced by our graph formulation/optimization method with the traditional level set approach. For both algorithms, the initialization, parameters and energy calculation method were identical.

4 Conclusion

In this work, we began by reformulating the classical Mumford-Shah energy functional in terms of analogous differential operators on graphs. An equivalent, alternate way of looking at this reformulation is to think of it as writing the MSF explicitly in terms of the specific finite operations that would be applied on a computer to estimate gradients and contour length. With this new reformulation, we may apply well-established combinatorial optimization techniques for producing reconstruction and contour updates.

Our experiments indicate a dramatic improvement of our graph-formulated optimization over a traditional contour evolution approaches. This improvement is in terms of speed, robustness to initialization, robustness to parameter settings and in production of a solution representing a lower MSF energy. Additionally, we employ combinatorial optimization that is not based on gradient descent to solve our graph formulation of the MSF, which permits non-local movement of the contour to find low energy solutions. It could be argued that several modifications to the level set method have been suggested for improving speed (e.g., multiresolution [4]) that were not employed in our experiments. Although we did not employ these modifications to the level set method, it seems unlikely that they would converge on real images as quickly as the proposed method. Even if it were possible to modify the contour evolution approach to converge after a few iterations to a good, low-energy solution, such an achievement would only serve to make the modified level set approach roughly equivalent in performance to the proposed approach, at the cost of additional complexity and optimization parameters. The contour evolution approach would still not permit non-local movement of the contour. Additionally, some modifications (e.g., multiresolution) could equally be applied to improve the performance of both approaches.

Finally, we hope that this work has illustrated the idea that a reformulation of traditional (continuous) PDE approaches in terms of their analogous differential operators on graphs (combinatorial operators) can permit the use of powerful combinatorial optimization techniques that may more quickly find lower energy solutions when compared to their standard level set counterparts. Although our primary motivation for reformulating traditionally continuous energies in terms of combinatorial operators is to provide faster, simpler, lower energy solutions capable of non-local movement of contours, it is important to note that a graph-based formulation permits application of the same techniques to more abstract domains, such as data clustering, mesh processing and space-variant vision.

References

1. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Comm. Pure and Appl. Math. 42, 577–685 (1989)
2. Chan, T., Vese, L.: Active contours without edges. IEEE TIP 10, 266–277 (2001)
3. Chan, T., Vese, L.: A level set algorithm for minimizing the Mumford-Shah functional in image processing. In: Workshop on VLSM, pp. 161–168. IEEE, Los Alamitos (2001)
4. Tsai, A., Yezzi, A., Willsky, A.: Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. IEEE TIP 10, 1169–1186 (2001)

5. Greig, D., Porteous, B., Seheult, A.: Exact maximum *a posteriori* estimation for binary images. Journal of the Royal Statistical Society, Series B 51, 271–279 (1989)
6. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE PAMI 26, 147–159 (2004)
7. Sinop, A.K., Grady, L.: A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In: Proc. of ICCV. IEEE, Los Alamitos (2007)
8. Boykov, Y., Kolmogorov, V., Cremers, D., Delong, A.: An integral solution to surface evolution PDEs via geo-cuts. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 409–422. Springer, Heidelberg (2006)
9. Zeng, X., Chen, W., Peng, Q.: Efficiently solving the piecewise constant Mumford-Shah model using graph cuts. Technical report, Zhejiang University (2006)
10. El-Zehiry, N., Xu, S., Sahoo, P., Elmaghraby, A.: Graph cut optimization for the Mumford-Shah model. In: Proc. of VIIP (2007)
11. Bougleux, S., Elmoataz, A., Melkemi, M.: Discrete regularization on weighted graphs for image and mesh filtering. In: SSVM, pp. 128–139. Springer, Heidelberg (2007)
12. Blake, A., Zisserman, A.: Visual Reconstruction. MIT Press, Cambridge (1987)
13. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. IEEE PAMI 6, 721–741 (1984)
14. Mattiussi, C.: The finite volume, finite element and finite difference methods as numerical methods for physical field problems. In: AIEP, pp. 1–146. Academic Press Inc., London (2000)
15. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: Proc. ICCV, vol. 1, pp. 26–33 (2003)
16. Grady, L., Schwartz, E.: Anisotropic interpolation on graphs: The combinatorial Dirichlet problem. Technical Report CAS/CNS-TR-03-014, Boston University (2003)
17. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE PAMI 26, 1124–1137 (2004)
18. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE PAMI 23, 1222–1239 (2001)
19. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In: Proc. of ICCV 2001, pp. 105–112 (2001)
20. Rother, C., Kolmogorov, V., Blake, A.: "GrabCut" — Interactive foreground extraction using iterated graph cuts. In: Proc. SIGGRAPH, vol. 23, pp. 309–314. ACM, New York (2004)
21. Sethian, J.: Level set methods and fast marching methods: evolving interfaces in computational geometry. Cambridge University Press, Cambridge (1999)
22. Osher, S., Sethian, J.: Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics 79, 12–49 (1988)
23. Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J., Osher, S.: Fast global minimization of the active contour/snake model. J. Mathematical Imaging and Vision 28, 151–167 (2007)
24. Wang, Z., Yang, X., Shi, P.: Solving Mumford-Shah model equation by AOS algorithm. In: Int. Conf. on Sig. Proc., vol. 1, pp. 740–743. IEEE Computer Society Press, Los Alamitos (2002)
25. Weickert, J., Romeny, B.M., Viergever, M.A.: Efficient and reliable schemes for nonliner diffusion filtering. IEEE Trans. on Image Proc. 7, 398–410 (1998)

Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features

Douglas Gray and Hai Tao

University of California, Santa Cruz
{dgray,tao}@soe.ucsc.edu
http://vision.soe.ucsc.edu/

Abstract. Viewpoint invariant pedestrian recognition is an important yet under-addressed problem in computer vision. This is likely due to the difficulty in matching two objects with unknown viewpoint and pose. This paper presents a method of performing viewpoint invariant pedestrian recognition using an efficiently and intelligently designed object representation, the ensemble of localized features (ELF). Instead of designing a specific feature by hand to solve the problem, we define a feature space using our intuition about the problem and let a machine learning algorithm find the best representation. We show how both an object class specific representation and a discriminative recognition model can be learned using the AdaBoost algorithm. This approach allows many different kinds of simple features to be combined into a single similarity function. The method is evaluated using a viewpoint invariant pedestrian recognition dataset and the results are shown to be superior to all previous benchmarks for both recognition and reacquisition of pedestrians.

1 Introduction

Pedestrian tracking is a deceptively hard problem. When the camera is fixed and the number of targets is small, pedestrians can easily be tracked using simple naive methods based on target location and velocity. However, as the number of targets grows, occlusion creates ambiguity. This can be overcome by delaying decisions and considering multiple hypothesis [1] and efficient solutions exist for solving this correspondence problem [2]. However, as the size of the scene itself grows, additional cameras are needed to provide adequate coverage. This creates another problem known as *pedestrian re-identification*. This is a much more challenging problem because of the lack of hard temporal (frame to frame) constraints when matching across non overlapping fields of view in a camera network. However, the ultimate goal of any surveillance system is not to track and reacquire targets, but to understand the scene and provide a more effective interface to the operator. Central to this goal is the ability to search the camera network for a person of interest. This is effectively the same as pedestrian re-identification without any temporal constraints. This problem of *pedestrian recognition* is the main subject of this paper.

Pedestrian recognition presents a number of challenges beyond the tracking problem, most importantly a lack of temporal information. Thus the matching

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 262–275, 2008.

decision must be made on the appearance model alone. So what is the best appearance model for pedestrian recognition?

The default option is a simple template, but this representation is viewpoint and pose specific. If the viewpoint angle is known one can compensate using flexible matching and alignment [3] [4] [5], but this will not work well for non-rigid objects. If the problem is limited to a frontal viewpoint then one could fit a triangular graph model [6] or part based model [7] to account for pose change. If multiple overlapping cameras are available it is possible to build a panoramic appearance map [8]. While template methods model the spatial layout of the object, histogram methods model its statistical properties. Histograms have proven useful for an assortment of tasks including tracking [9], texture classification [10], and pedestrian detection [11]. Many attempts have been made to combine the advantages of templates and histograms. Past approaches include recording correlations in correlograms [12], spatial position in spatiograms [13], vertical position in principal axes [14], or scale in multi-resolution histograms [15]. Both template and histogram methods suffer from problems with illumination changes, however it has been shown that this can be compensated for by learning the brightness transfer function between cameras [16].

The appearance model presented in this paper is a hybrid of the template and histogram, however instead of designing the model by hand, machine learning is used to construct a model that provides maximum discriminability for a set of training data. The learned model is an ensemble of localized features, each consisting of a feature channel, location and binning information, and a likelihood ratio test for comparing corresponding features. Once the model has been learned, it provides a similarity function for comparing pairs of pedestrian images. This function can be used for both pedestrian re-identification and recognition. In a practical implementation of the latter case it is expected that

Fig. 1. Some examples from the viewpoint invariant pedestrian recognition (VIPeR) dataset [17]. Each column is one of 632 same-person example pairs.

a human operator would be involved, so we provide both the recognition rate and the expected search time for a human operator.

While there is a great deal of data available for pedestrian detection, training a similarity function for recognition requires multiple images of the same individual. We have chosen to use the VIPeR dataset [17], which contains two views of 632 pedestrians. Some examples of pedestrian image pairs can be found in figure 1. Results for the proposed method are presented in section 4 and shown to far exceed the existing benchmarks.

2 Learning the Similarity Function

Learning domain specific distance or similarity functions is an emerging topic in computer vision [18] [19] [20] [21]. Some have attempted to learn fast approximations [20] to other more computationally expensive functions such as EMD [22]. While others have focused on the features that are found in the process [19]. These approaches can be summarized as follows: AdaBoost is used to sequentially learn computationally inexpensive features to solve a classification problem. Our approach is quite similar in these respects, however our similarity function is domain specific (*i.e.* only applicable to comparing pedestrian images).

The proposed similarity function is a weighted ensemble of likelihood ratio tests, constructed with the AdaBoost algorithm, a brief review of which can be found in Algorithm 1. At each iteration of the algorithm, the feature space is searched for

Algorithm 1: AdaBoost

Given:

- N labeled example training examples (x_i, y_i) where x_i is a pair of pedestrian images and $y_i \in \{-1, 1\}$ denotes if the two image are of the same person.
- A distribution over all training examples: $D_1(i) = 1/N$ for $i = 1 \dots N$.

For $t = 1, \dots, T$:

- Find the best localized feature λ_t and model m_t for the current distribution D_t.
- Calculate the edge γ_t

$$\gamma_t = \sum_{i=1}^{N} D_t(i) h(x_i) y_i$$

- If $\gamma_t < 0$ break
- Set $\alpha_t = \frac{1}{2} \ln \frac{1+\gamma_t}{1-\gamma_t}$
- Set $D_{t+1}(i) = \frac{1}{Z_t} D_t(i) \exp\left(-\alpha_t h(x_i) y_{(i)}\right)$, where Z_t is a normalizing factor
- Add α_t, λ_t and m_t to the ensemble

Output the ensemble of weights as A, features as Λ, and models as M.

Fig. 2. The AdaBoost algorithm for learning the similarity function

the best classifier w.r.t. the current distribution and added to the ensemble. In order to keep the problem tractable, we have selected a feature space that can be searched in a reasonable amount of time and is appropriate for the class of input data. While the main objective is to learn an effective similarity function, the size of the classifier is also important. For this reason the input to each classifier is always selected to be a scalar.

2.1 Formulation

The proposed task is to simultaneously learn a set of discriminative features and an ensemble of classifiers. We begin with the following set of definitions. The set of features learned is defined as $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_T\}$. Each feature consists of three elements: a feature channel, a region, and a histogram bin. Denoted as:

$$\lambda =< \text{channel}, (x, y, w, h), (\min, \max) > \tag{1}$$

A specific instance of a pedestrian is defined as $\mathbf{V} = [v_1, v_2, \ldots, v_T]^{\mathrm{T}}$ where each $v_i = p(\lambda_i|I)$, is the probability of a pixel from the specified channel and region being in the specified range. The set of models used to discriminate between two specific instances is defined as $M = \{m_1, m_2, \ldots, m_T\}$, where each m_i denotes the parameters of a likelihood ratio test.

2.2 Feature Channels

A feature channel is any single channel transformation of the original image. Two varieties of feature channel are explored in this paper, color and texture. Eight color channels corresponding to the three separate channels of the RGB YCbCr and HSV[1] colorspace are considered, as well as nineteen texture channels. Two families of texture filters are used, Schmid [23] and Gabor [24]. Each texture channel is the result of convolution with a filter and the luminance channel.

Schmid filters are defined here as:

$$F(r, \sigma, \tau) = \frac{1}{Z} \cos\left(\frac{2\pi\tau r}{\sigma}\right) e^{-\frac{r^2}{2\sigma^2}} \tag{2}$$

Where r denotes the radius, Z is a normalizing constant, and the parameters τ and σ are set to (2,1), (4,1), (4,2), (6,1), (6,2), (6,3), (8,1), (8,2), (8,3), (10,1), (10,2), (10,3), and (10,4) respectively. These filters were originally designed to model rotation invariant texture, but their use here is motivated by the desire to be invariant to viewpoint and pose. Additionally, six Gabor filters are used with parameters set to γ, θ, λ and σ^2 set to (0.3,0,4,2), (0.3,0,8,2), (0.4,0,4,1), (0.4,0,4,1), (0.3,$\frac{\pi}{2}$,4,2), (0.3,$\frac{\pi}{2}$,8,2), (0.4,$\frac{\pi}{2}$,4,1) and (0.4,$\frac{\pi}{2}$,4,1) respectively. All 19 texture filters used here can be seen in figure 3.

Other filters could be added as well, but proved less effective. It has been observed that adding additional features has few drawbacks other than increasing computational and storage requirements. The methodology used to select these specific channels was somewhat haphazard, so it is likely that better feature channels may still be found.

[1] Only one of the luminance (Y and V) channels is used.

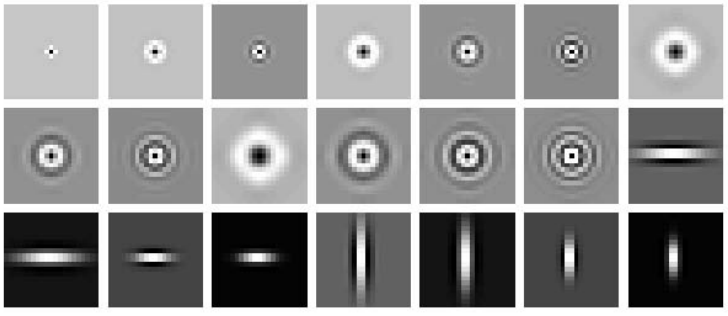


Fig. 3. The filters used in the model to describe texture. (a) Rotationally symmetric Schmid filters. (b) Horizontal and vertical Gabor filters.

2.3 Feature Regions

A feature region could be any collection of pixels in the image, but for reasons of computational sanity they will be restricted to a more tractable subset. Some popular subsets of regions include the simple rectangle, a collection of rectangles [19], or a rectangularly shaped region [7]. The motivation for this has been the computational savings of computing sums over rectangular regions using an integral image. However we can use our intuition about the problem to significantly reduce the number of regions to be considered. Since we know the data consists of pedestrians seen from an arbitrary horizontal viewpoints, we can disregard the horizontal dimension as it is not likely to be relevant, which leaves us with a set of "strips", or rectangles which span the entire horizontal dimension.

2.4 Feature Binning

A feature bin is simply a range over which pixel values are counted. In a traditional histogram, an orthogonal collection of bins is selected to uniformly cover the range of possible values. While this is justified in the general case where the image domain and task are unknown, there is little justification for this approach here, as both computation time and storage can be saved by selecting only the regions of bin space that are relevant to discriminating between pedestrians.

2.5 Feature Modeling

The basis for our similarity function is a collection of likelihood ratio tests performed on the features $\mathbf{V}$. Each test is performed on the value δ, which is defined as the absolute difference between two instances of the same feature:

$$\delta = |v^{(a)} - v^{(b)}| \tag{3}$$

The training data for the proposed approach consists of a collection of pedestrian images. Each individual is seen from two different camera angles, denoted (a) and

(b). δ is computed for every pair of training images between the two cameras. If there are N individuals, then there are N positive training examples denoted Δ_{p}, and $N(N-1)$ negative training examples denoted Δ_{n}. Three possible probability distributions are considered here, Exponential, Gamma, and Gaussian. For each model, the parameters are estimated and a likelihood ratio is computed. This gives three possible weak classifiers for the ensemble.

If δ is distributed as exponential:

$$h(v^{(a)}, v^{(b)}) = \begin{cases} 1 & \text{If } a\delta + b > 0 \\ -1 & \text{Otherwise} \end{cases} \tag{4}$$

Where the coefficients a and b can be expressed in terms of the estimated parameters of the positive and negative distributions as:

$$a = \widehat{\lambda_n} - \widehat{\lambda_p} \qquad b = \ln(\widehat{\lambda_p}) - \ln(\widehat{\lambda_n}) \tag{5}$$

The parameters of an exponential distribution can be estimated as $\widehat{\lambda} = \frac{1}{\mu}$.

If δ is distributed as gamma:

$$h(v^{(a)}, v^{(b)}) = \begin{cases} 1 & \text{If } a\delta + b\ln\delta + c > 0 \\ -1 & \text{Otherwise} \end{cases} \tag{6}$$

Where the coefficients a, b and c can be expressed in terms of the estimated parameters of the positive and negative distributions as:

$$\begin{gathered} a = \hat{\beta}_n - \hat{\beta}_p \qquad b = \hat{\alpha_p} - \hat{\alpha_n} \\ c = \hat{\alpha_p}\ln\hat{\beta}_p - \hat{\alpha_n}\ln\hat{\beta}_n + \ln\Gamma(\hat{\alpha_n}) - \ln(\Gamma\hat{\alpha_p}) \end{gathered} \tag{7}$$

The parameters of a gamma distribution can be estimated as $\hat{\alpha} = \frac{\hat{\mu}^2}{\hat{\sigma}^2}$ and $\hat{\beta} = \frac{\hat{\mu}}{\hat{\sigma}^2}$.

If δ is distributed as Gaussian:

$$h(v^{(a)}, v^{(b)}) = \begin{cases} 1 & \text{If } a\delta^2 + b\delta + c > 0 \\ -1 & \text{Otherwise} \end{cases} \tag{8}$$

Where the coefficients a, b and c can be expressed in terms of the estimated parameters of the positive and negative distributions as:

$$\begin{gathered} a = \frac{1}{2\hat{\sigma_n^2}} - \frac{1}{2\hat{\sigma_p^2}} \qquad b = \frac{\hat{\mu_p}}{\hat{\sigma_p^2}} - \frac{\hat{\mu_n}}{\hat{\sigma_n^2}} \\ c = \frac{\hat{\mu_n^2}}{2\hat{\sigma_n^2}} - \frac{\hat{\mu_p^2}}{2\hat{\sigma_p^2}} + \ln(\hat{\sigma_n^2}) - \ln(\hat{\sigma_p^2}) - \ln(2\pi) \end{gathered} \tag{9}$$

2.6 Search Strategy

At each iteration we must find the best feature for the current distribution. The traditional approach is to define the feature space to be small enough that an exauhstive search is possible at each iteration. The size of the feature space proposed here is the product of the number of possible channels, regions, bins, and models. While the number of possible channels and models is relatively

small (21 and 3 respectively), the number of possible contiguous regions and bins grows quadratically with the number of quantization levels. Thus without any quantization, the total search space would be $|\mathcal{F}| \approx 10^{10}$.

We have found the following steps greatly improved training time. First, we precompute an intermediate feature representation for every image before training. This feature is a quantized two dimensional map of each channel of each image. The two dimensions of this map are the quantized y coordinate and pixel value. This map is then transformed into an integral image, allowing for any histogram bin to be calculated over any set of vertical strips in constant time using the integral image trick. Second, a coarse to fine search strategy is used to explore the parts of the feature space that we believe to be smooth (*e.g.* the region and binning space). As a result of these search strategies the search time has been reduced from hours to minutes.

2.7 What Is the Model Being Learned?

The usual approach to solving a problem such as this is for the researcher to hand craft a feature representation that appears appropriate for the class of data and then select the distance function or classifier that provides the best results. For example Park *et al.* noticed that people often wear different color shirt and pants, and thus defined their feature representation to be three histograms taken over the head, shirt and pants regions of a person [25]. Hu *et al.* noticed that the principal axis is often different among different pedestrians and choose a model accordingly [14]. Gheissari *et al.* have taken the extreme approach of designing a 2d mesh model of a pedestrian in order to obtain an exact correspondence between frontal pedestrian images [6].

As researchers we never really know what the *correct* model to use in any particular problem is. However we have a great deal of intuition about how we as humans would solve the problem. What we have done here is use our intuition to define a broad (*i.e.* intentionally vague) feature space that we believe contains a good feature representation, and then allowed the AdaBoost algorithm to build the best model for the training data available. In this case, the model is a collection of simple color and texture features and some spatial and intensity information.

3 Modeling Pedestrian Recognition Performance

3.1 Evaluation Data

The experimental setups used to evaluate pedestrian recognition have varied widely. Gandhi and Trivedi provide results on 10 individuals seen simultaneously from 4 opposing viewpoints which are used to create a panoramic image map [8]. This setup is ideal for tracking, but very costly to implement. Gheissari et al. collected 3 frontal views of 44 unique individuals [6] and Wang et al. later added 99 additional individuals to that dataset [7]. While their data contains a diverse set of people and poses, the lack of viewpoint variation is insufficient to

model the real world recognition problem. Gray et al. have collected two views of 632 individuals seen from widely differing viewpoints [17]. In contrast to the aforementioned data, *most* of their examples contain a viewpoint change of 90 degrees or more, making recognition very challenging. The method presented in this paper is evaluated using their public dataset. Some examples from this dataset can be found in figure 1.

3.2 Evaluation Methodology

Several approaches have been used for evaluating recognition and re-identification performance. Shan *et al.* has treated the re-identification problem as a same-different detection problem and provided results using a receiver operating characteristic (ROC) curve [4]. Wang *et al.* treat the problem as recognition and provide results using a cumulative matching characteristic (CMC) curve. Gray *et al.* provide results using a CMC curve, but also present a method of converting their results into a re-identification rate. This paper presents results in the form of a recognition rate (CMC), re-identification rate, and expected search time by a human operator.

The evaluation methodology used here is as follows. The training data is split evenly into a training and test set. Each image pair is split and randomly assigned to camera a and camera b. The CMC curve for the test set is calculated by selecting a probe (image from camera a) and matched with a gallery (every image in camera b). This provides an ranking for every image in the gallery w.r.t the probe. This procedure is repeated for every image in camera a and averaged. Camera a and camera b are then swapped and the process is repeated. The CMC curve is then the expectation of finding the correct match in the top n matches.

The expected search time is defined as the expected amount of time required for a human operator to find the correct match from a collection of images. If we assume the human operator makes no mistakes, then we can decomposed the expected search time into three components:

$$\text{E[Search Time]} = \frac{\text{E[Sort Position]}}{\text{Dataset Size}} \times \text{E[Time per Image]} \times \text{Dataset Size}$$

If the system has no prior knowledge about the probable association between the probe and gallery images, then the images will be presented in random order and the first term will be 0.5. For the sake of simplicity, we will assume the second term is one second, making the expected search time and sort position the same value.

Thus there are three ways to reduce the expected search time. The human operator could take less time per image at the expense of missing the correct match. The size of the dataset could be reduced using other information (eg. spatial or temporal information about camera position). Or the operator could be presented with the data sorted by some similarity function. This reduction in search time may be small when the dataset size is small (ie. in laboratory testing), however in a real world scenario this number could be quite large and the time savings could be very significant.

4 Results

4.1 Benchmarks

We compare our results to 4 different benchmark methods. A simple template (SSD matching), a histogram, a hand localized histogram with the configuration proposed by Park *et al.* [25], and a principal axis histogram, which is similar in spirit to the work of Hu *et al.* [14]. Multiple configurations were tried for each method, but only the best results for each are shown here. We found that 16 quantization levels, YCbCr colorspace and the Bhattacharyya distance performed best for all three histogram methods. The key differences between the three approaches are the regions over which histograms are computed. In the hand localized histogram, three regions are chosen to correspond to the head (top $\frac{1}{5}$), shirt (middle $\frac{2}{5}$) and pants (bottom $\frac{2}{5}$)). In the principal axis histogram 32 regions are chosen in 4 pixel increments to cover each horizontal stripe of the image. The ELF model shown here contains 200 features and will be analyzed in greater detail in section 4.4.

Comparisons with the methods proposed in [6] and [7] are desirable, but not practical given the complexity of these methods. Additionally, these two methods were designed for frontal viewpoints, and would likely give poor results on the data used here because of the wide viewpoint changes.

4.2 Recognition

As was mentioned in the beginning of this paper, pedestrian recognition is a hard problem. Figure 4 shows an example of 16 probe images, the top 28 matches using our similarity function, and the correct match. Given a single image, finding its counterpart from a gallery of 316 images is quite challenging for a human operator. We challenge the reader to find the correct matches in this sorted gallery without looking at the key in the caption or the corresponding image on the right.

Figure 5 shows recognition performance as a CMC curve. The rank 1 matching rate of our approach is around 12%, while the correct match can be found in the top 10% (rank 31) around 70% of the time. The utility of these recognition rates can be summarized by looking at the expected search times in figure 6. Without any similarity measure, a user would have to look at half the data on average before finding the correct match. This could take quite some time for a large number of images. At this task, the ELF similarity function yields an 81.7% search time reduction over a pure human operator, and a 58.2% reduction over the next best result.

4.3 Re-identification

The difficulty in pedestrian re-identification varies with the number of possible targets to match. Figure 7 shows how the re-identification performance of the different approaches performs as the number of possible targets increases.

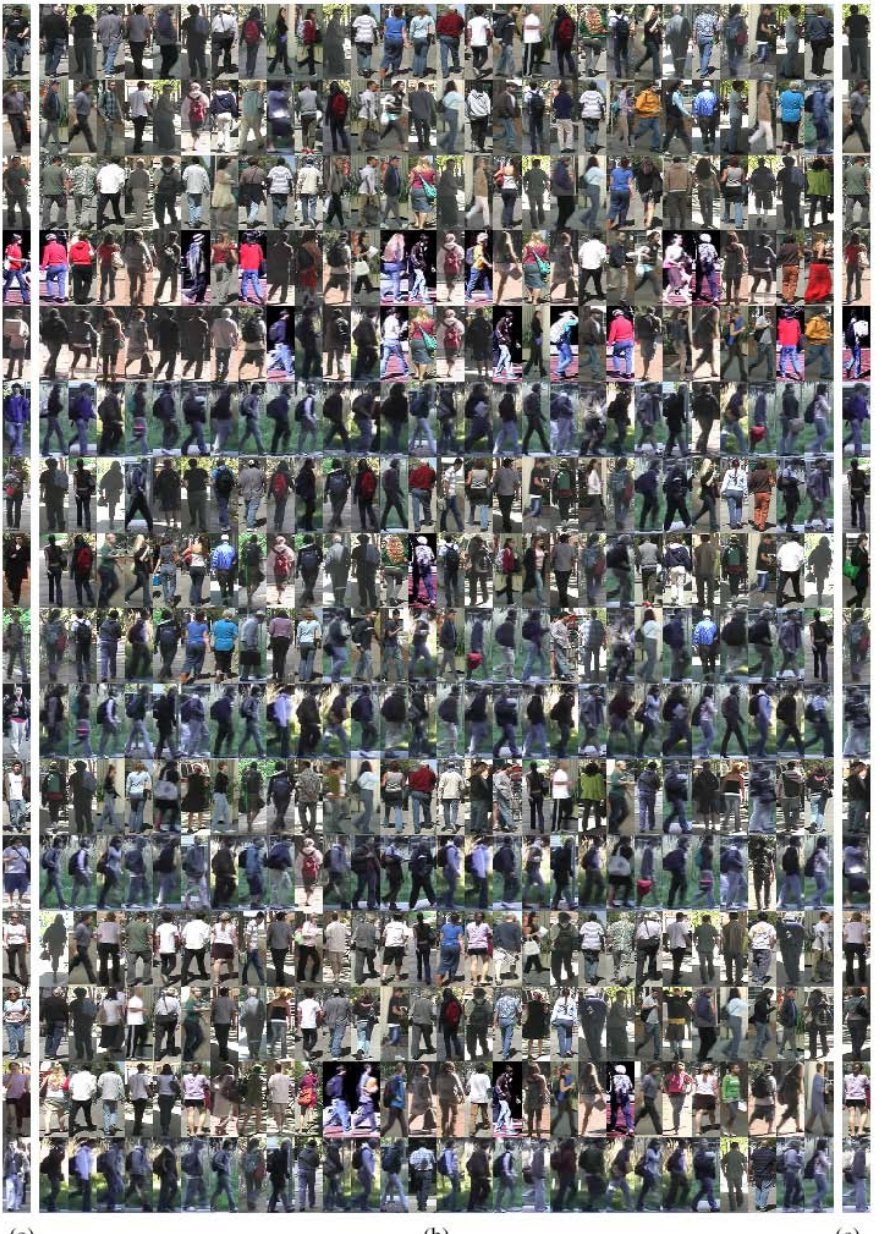


Fig. 4. Example queries to a recognition database. (a) Probe image. (b) Top n results (sorted left to right). (c) Correct match. Note the visual similarity of the returned results and ambiguity created by pose, viewpoint and lighting variations. The correct match for these examples was ranked 2, 2, 1, 3, 39, 2, 2, 196, 1, 33, 16, 55, 3, 45, 6 and 18 respectively from a gallery of 316 people (top to bottom).

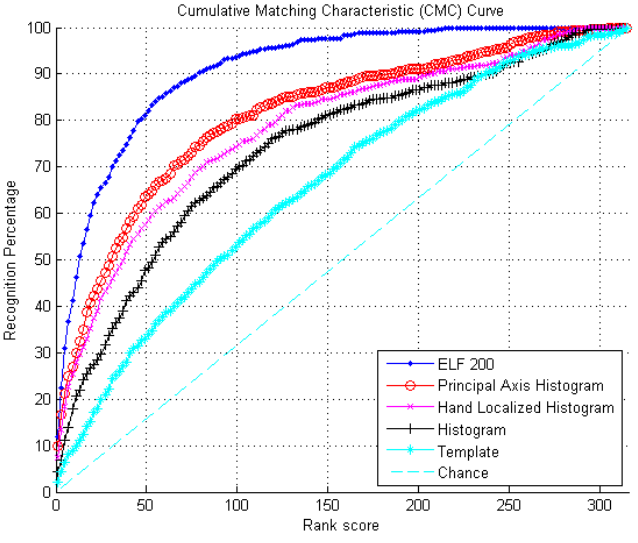


Fig. 5. Cumulative matching characteristic (CMC) curve for ELF model and benchmarks

Method	Expected Search Time (s)
Chance	158.0
Template	109.0
Histogram	82.9
Hand Localized Histogram	69.2
Principal Axis Histogram	59.8
ELF	28.9

Fig. 6. Expected search time for ELF model and benchmarks. This assumes a human operator can review 1 image per second at 100% accuracy.

When the number of possible targets is small, performance is very high. The re-identification task is rarely performed with appearance models alone. In most indoor or otherwise restricted camera networks, spatial information can be used to restrict the number of matches to be quite small, making these results very promising considering that spatial and temporal information can be combined with appearance information to great effect as these cues are independent.

4.4 Model Analysis

One of the strengths of this approach is the ability to combine many different kinds of features into one similarity function. Figure 8 shows the percentage of weight accorded to each feature channel or family of features. It is not surprising given the illumination changes between the two cameras, that the two most informative channels are hue and saturation. Roughly three quarters of the weight

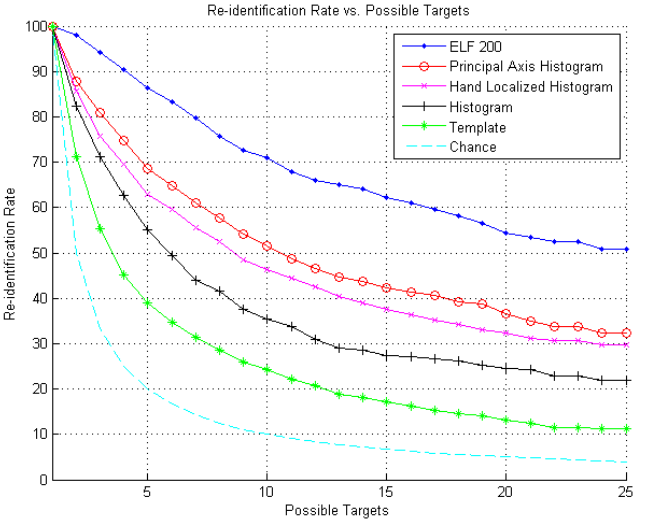


Fig. 7. Re-identification rate *vs.* number of targets for ELF model and benchmarks

Feature Channel	Percent of classifier weight
R	11.0 %
G	9.4 %
B	12.4 %
Y	6.4 %
Cb	6.1 %
Cr	4.5 %
H	14.2 %
S	12.5 %
Schmid	12.0 %
Gabor	11.7 %

Fig. 8. A table showing the percent of features from each channel, model

of the classifier is devoted to color features, which seems to suggest that past approaches which relied on color histograms alone were justified.

5 Conclusions

We have presented a novel approach to viewpoint invariant pedestrian recognition that learns a similarity function from a set of training data. It has been shown that this ensemble of localized features is effective at discriminating between pedestrians regardless of the viewpoint change between the two views. While the automatic pedestrian recognition problem remains unsolved, it has been shown that the proposed approach can be used to assist a human operator in this task by significantly reducing the search time required to match

pedestrians from a large gallery. While the ultimate goal of automated surveillance research is to remove the human entirely, this work represents a significant improvement over past approaches in reducing the time required to complete simple surveillance tasks such as recognition and re-identification.

References

1. Reid, D.: An algorithm for tracking multiple targets. Automatic Control, IEEE Transactions on 24(6), 843–854 (1979)
2. Cox, I., Hingorani, S., et al.: An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(2), 138–150 (1996)
3. Guo, Y., Hsu, S., Shan, Y., Sawhney, H.: Vehicle fingerprinting for reacquisition & tracking in videos. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2 (2005)
4. Shan, Y., Sawhney, H., Kumar, R.: Vehicle Identification between Non-Overlapping Cameras without Direct Feature Matching. In: IEEE International Conference on Computer Vision, vol. 1 (2005)
5. Guo, Y., Shan, Y., Sawhney, H., Kumar, R.: PEET: Prototype Embedding and Embedding Transition for Matching Vehicles over Disparate Viewpoints. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2007)
6. Gheissari, N., Sebastian, T., Hartley, R.: Person Reidentification Using Spatiotemporal Appearance. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1528–1535 (2006)
7. Wang, X., Doretto, G., Sebastian, T., Rittscher, J., Tu, P.: Shape and appearance context modeling. In: IEEE International Conference on Computer Vision, pp. 1–8 (2007)
8. Gandhi, T., Trivedi, M.: Person tracking and reidentification: Introducing Panoramic Appearance Map (PAM) for feature representation. Machine Vision and Applications 18(3), 207–220 (2007)
9. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2 (2000)
10. Varma, M., Zisserman, A.: A Statistical Approach to Texture Classification from Single Images. International Journal of Computer Vision 62(1), 61–81 (2005)
11. Dalai, N., Triggs, B., Rhone-Alps, I., Montbonnot, F.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1 (2005)
12. Huang, J., Ravi Kumar, S., Mitra, M., Zhu, W., Zabih, R.: Spatial Color Indexing and Applications. International Journal of Computer Vision 35(3), 245–268 (1999)
13. Birchfield, S., Rangarajan, S.: Spatiograms versus Histograms for Region-Based Tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2 (2005)
14. Hu, W., Hu, M., Zhou, X., Lou, J.: Principal Axis-Based Correspondence between Multiple Cameras for People Tracking. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28(4) (2006)
15. Hadjidemetriou, E., Grossberg, M., Nayar, S.: Spatial information in multiresolution histograms. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 702–709 (2001)

16. Javed, O., Shafique, K., Shah, M.: Appearance Modeling for Tracking in Multiple Non-overlapping Cameras. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 26–33 (2005)
17. Gray, D., Brennan, S., Tao, H.: Evaluating Appearance Models for Recognition, Reacquisition, and Tracking. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS) (2007)
18. Hertz, T., Bar-Hillel, A., Weinshall, D.: Learning distance functions for image retrieval. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2 (2004)
19. Dollar, P., Tu, Z., Tao, H., Belongie, S.: Feature Mining for Image Classification. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2007)
20. Athitsos, V., Alon, J., Sclaroff, S., Kollios, G.: Boostmap: An embedding method for efficient nearest neighbor retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(1), 89–104 (2008)
21. Yu, J., Amores, J., Sebe, N., Radeva, P., Tian, Q.: Distance learning for similarity estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(3), 451–462 (2008)
22. Rubner, Y., Tomasi, C., Guibas, L.: The Earth Mover's Distance as a Metric for Image Retrieval. International Journal of Computer Vision 40(2), 99–121 (2000)
23. Schmid, C.: Constructing models for content-based image retrieval. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2 (2001)
24. Fogel, I., Sagi, D.: Gabor filters as texture discriminator. Biological Cybernetics 61(2), 103–113 (1989)
25. Park, U., Jain, A., Kitahara, I., Kogure, K., Hagita, N.: ViSE: Visual Search Engine Using Multiple Networked Cameras. In: IEEE International Conference on Pattern Recognition, 1204–1207 (2006)

Perspective Nonrigid Shape and Motion Recovery

Richard Hartley[1] and René Vidal[2]

[1] Australian National University and NICTA, Canberra, ACT, Australia
[2] Center for Imaging Science, Johns Hopkins University, Baltimore, MD, USA

Abstract. We present a closed form solution to the nonrigid shape and motion (NRSM) problem from point correspondences in multiple perspective uncalibrated views. Under the assumption that the nonrigid object deforms as a linear combination of K rigid shapes, we show that the NRSM problem can be viewed as a reconstruction problem from multiple projections from $\mathbb{P}^{3K}$ to $\mathbb{P}^2$. Therefore, one can linearly solve for the projection matrices by factorizing a multifocal tensor. However, this projective reconstruction in $\mathbb{P}^{3K}$ does not satisfy the constraints of the NRSM problem, because it is computed only up to a projective transformation in $\mathbb{P}^{3K}$. Our key contribution is to show that, by exploiting algebraic dependencies among the entries of the projection matrices, one can upgrade the projective reconstruction to determine the affine configuration of the points in $\mathbb{R}^3$, and the motion of the camera relative to their centroid. Moreover, if $K \geq 2$, then either by using calibrated cameras, or by assuming a camera with fixed internal parameters, it is possible to compute the Euclidean structure by a closed form method.

1 Introduction

Structure from motion (SfM) refers to the problem of reconstructing a 3-D rigid scene from multiple 2-D images taken by a moving camera. This is a well studied problem in computer vision (see for instance [1,2]), which has found numerous applications in image-based modeling, human-computer interaction, robot navigation, vision-based control, etc.

A fundamental limitation of classical SfM algorithms is that they cannot be applied to scenes containing nonrigid objects, such as scenes containing articulated motions, facial expressions, hand gestures, etc. This has motivated the development of a family of methods where a moving *affine calibrated* camera observes a nonrigid shape that deforms as a linear combination of K rigid shapes with time varying coefficients [3,4,5,6,7,8]. This assumption allows one to recover nonrigid shape and motion (NRSM) using extensions of the classical rigid factorization algorithm of Tomasi and Kanade [9]. For instance, Bregler et al. [5] use multiple matrix factorizations to enforce orthonormality constraints on camera rotations. Brand [3] uses a non-linear optimization method called flexible factorization. Torresani et al. [7] use a trilinear optimization algorithm that alternates between the computation of shape bases, shape coefficients, and camera rotations. Xiao et al. [8] provide a characterization of the space of ambiguous solutions as well as a closed form solution by enforcing additional *shape constraints* on the shape bases. Their solution not only applies to shapes of full rank three, but can also be extended to degenerate rank one and two shapes, as shown in [10].

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 276–289, 2008.

An important assumption made by these approaches is that the projection model is *affine* and the camera is *calibrated*. One way of extending affine methods to the projective case is to alternate between the estimation of the projective depths and the estimation of shape and motion, similarly to the Sturm and Triggs algorithm [11]. This approach was indeed explored in [12] for the NRSM problem. However, it is well known that iterative schemes are often very sensitive to initialization. In the rigid case the projective depths can be initialized using algebraic methods based on two-view geometry. In the nonrigid case, the situation is obviously not as straightforward, and hence the method of [12] simply assumes the initial depths to be all equal to one. To the best of our knowledge, the only existing algebraic solution to the perspective NRSM problem can be found in [13], where it is shown that the problem is solvable for a number of views F in the range $(3K+1)/2 \leq F \leq (3K+1)$. However, the algorithm for computing shape and motion relies on the factorization of a quintifocal tensor, and is applicable only in the case of two shape bases seen in five calibrated perspective views.

In this paper, we present a closed form solution to nonrigid shape and motion recovery for an arbitrary number of shape bases K and an arbitrary number F of perspective uncalibrated views in the range $(3K+1)/2 \leq F \leq (3K+1)$. Our solution exploits the fact that the NRSM problem can be viewed as a reconstruction problem from $\mathbb{P}^{3K}$ to $\mathbb{P}^2$ where the projection matrices have a particular structure. As shown in [14], the camera projections associated with any reconstruction problem from $\mathbb{P}^n$ to $\mathbb{P}^m$ can be computed in closed form from the factorization of a multifocal tensor. However, the projection matrices computed by this method do not necessarily conform with the particular structure of the NRSM problem, because they are computed up to a projective transformation in $\mathbb{P}^{3K}$ only. The main contribution of our work is to show that one can solve for the projective transformation, and hence for the camera matrices, shape basis, and shape coefficients, in closed form using linear algebraic techniques which do not require the use of iteration. More specifically, we show that the NRSM problem can be solved as follows:

1. Linearly compute a multifocal tensor from point correspondences in multiple views of a nonrigid object.
2. Factorize the multifocal tensor into $\mathbb{P}^{3K} \rightarrow \mathbb{P}^2$ projection matrices, defined up to a common projective transformation of $\mathbb{P}^{3K}$.
3. Compute a normalizing projective transformation by enforcing internal constraints on the projection matrices.
4. Compute the camera matrices, shape basis and shape coefficients from the normalized projection matrices.

Using this method, we find the following results, when the number of shape bases is $K \geq 2$.

1. The structure of the point set may be determined in each frame, up to an affine transformation common to all frames. This is in contrast with the classic reconstruction problem with a single shape basis, where the structure may be computed only up to a projective transformation.
2. If the cameras are calibrated, or have constant internal parameters, then the Euclidean shape may be determined by closed form or linear techniques.

3. Since the points are potentially moving (within the space spanned by the K shape bases), it is possible to determine the camera motion only relative to the moving points, and up to an individual scaling in each frame. This is the only ambiguity of the reconstruction (other than a choice of the affine or Euclidean coordinate frame). If the points are assumed to be centred at the origin, then the camera motion is uniquely determined apart from a scale within the affine or Euclidean coordinate frame.

Paper Contributions. This paper gives the first non-iterative solution for the general nonrigid perspective structure-from-motion problem. Because of the deterministic nature of the algorithm, it is guaranteed to find the correct solution at least for noise-free data. This is not the case with previous iterative algorithms. (For the difficulties involved with such iterative methods, see for instance [15].) Further, our analysis allows us to discover the fundamental ambiguities and limitations of NRSM, both in the affine and perspective cases. Our results clarify and complete the previous results on the ambiguities of affine NRSM given in [16,13].

2 Nonrigid Shape and Motion Problem

Notation. We make extensive use of the Kronecker or tensor product $\mathtt{A} \otimes \mathtt{B}$, where $\mathtt{A}$ and $\mathtt{B}$ are matrices. This tensor product is given by

$$\mathtt{A} \otimes \mathtt{B} = \begin{bmatrix} a_{11}\mathtt{B} & \dots & a_{1n}\mathtt{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathtt{B} & \dots & a_{mn}\mathtt{B} \end{bmatrix},$$

where the a_{ij} are the elements of $\mathtt{A}$. A basic property is that $(\mathtt{A}\otimes\mathtt{B})\,(\mathtt{C}\otimes\mathtt{D}) = (\mathtt{AC})\otimes(\mathtt{BD})$ whenever the dimensions are compatible so that this equality makes sense. Consequently, if $\mathtt{A}$ and $\mathtt{B}$ are square, then $(\mathtt{A} \otimes \mathtt{B})^{-1} = \mathtt{A}^{-1} \otimes \mathtt{B}^{-1}$.

We use the notation $\mathrm{stack}(\dots)$ to represent the matrix or vector created by stacking its arguments (matrices or vectors) vertically.

Bold font ($\mathbf{X}, \mathbf{x}$) is used to represent vectors (one-dimensional arrays) and typewriter font ($\mathtt{A}, \mathtt{W}, \dots$) to represent matrices (two-dimensional arrays). Given a homogeneous vector, such as $\mathbf{x}$ or $\mathbf{X}$, the corresponding non-homogeneous vector is denoted with a hat, such as $\hat{\mathbf{x}}$ or $\widehat{\mathbf{X}}$. Notation such as $\Pi_{a:b}$ represents rows a to b of Π.

Finally, for inline representation of simple matrices, we use the notation $[a, b\,;\, c, d]$, where the elements are listed in row major order, rows separated by a semi-colon.

Problem statement. Let $\{\mathbf{x}_{fp} \in \mathbb{P}^2 \mid p = 1, \dots, P\,;\; f = 1, \dots, F\}$ be the perspective projections of P (possibly moving) 3-D points $\{\mathbf{X}_{fp} \in \mathbb{P}^3\}$ onto F frames from a moving camera. Let $\mathtt{P}_f = \begin{bmatrix}\mathtt{M}_f & \mathbf{t}_f\end{bmatrix} \in \mathbb{R}^{3\times 4}$ be the *camera matrix* associated with frame f. Then

$$\lambda_{fp}\mathbf{x}_{fp} = \mathtt{P}_f\mathbf{X}_{fp}, \tag{1}$$

where λ_{fp} is an unknown scale factor, called *projective depth*. It follows that

$$\mathtt{W} = \begin{bmatrix} \lambda_{11}\mathbf{x}_{11} \cdots \lambda_{1P}\mathbf{x}_{1P} \\ \vdots \quad\quad \vdots \\ \lambda_{F1}\mathbf{x}_{F1} \cdots \lambda_{FP}\mathbf{x}_{FP} \end{bmatrix} = \begin{bmatrix} \mathtt{P}_1\mathtt{X}_1 \\ \vdots \\ \mathtt{P}_F\mathtt{X}_F \end{bmatrix}, \tag{2}$$

where $\mathtt{X}_f = \begin{bmatrix} \mathbf{X}_{f1} \; \mathbf{X}_{f2} \cdots \; \mathbf{X}_{fP} \end{bmatrix} \in \mathbb{R}^{4\times P}$ is called the *structure matrix* and is formed from the homogeneous coordinates of all the P points in the f-th frame.

The *structure from motion problem* (SfM) refers to the problem of recovering the camera matrices $\mathtt{P}_f$, and the structure matrices $\mathtt{X}_f$ from measurements of the image point trajectories $\mathbf{x}_{fp}$. Without some restriction on the moving 3-D points, the SfM problem is of course not solvable.

When the P points lie on a rigid stationary object, the structure matrices are equal, that is $\mathtt{X}_1 = \mathtt{X}_2 = \cdots = \mathtt{X}_F = \mathtt{X}$. Hence, given the depths one can factorize $\mathtt{W}$ into a motion matrix $\mathtt{\Pi} \in \mathbb{R}^{3F\times 4}$ and a structure matrix $\mathtt{X} \in \mathbb{R}^{4\times P}$ as $\mathtt{W} = \mathtt{\Pi X}$. This rank constraint has been the basis for all factorization-based algorithms, e.g. [9,11]. In fact, one can solve the SfM problem by alternating between the estimation of the depths, and the estimation of motion and structure [17], though care must be taken to avoid converging to trivial solutions [15].

In this paper we study the case where the 3-D points lie on a nonrigid object, thereby allowing the 3-D points $\mathbf{X}_{fp}$ to move as a function of time. As suggested in [3,4,5,6,7], we assume that the P points deform as a linear combination of a fixed set of K rigid shape bases with time varying coefficients. That is, $\widehat{\mathtt{X}}_f = \sum_{k=1}^{K} c_{fk}\widehat{\mathtt{B}}_k$, where the matrix $\widehat{\mathtt{X}}_f = [\widehat{\mathbf{X}}_{f1} \cdots \widehat{\mathbf{X}}_{fP}] \in \mathbb{R}^{3\times P}$ is the *object shape* at frame f, the matrices $\{\widehat{\mathtt{B}}_k = \begin{bmatrix} \widehat{\mathbf{B}}_{k1} \cdots \widehat{\mathbf{B}}_{kP} \end{bmatrix} \in \mathbb{R}^{3\times P}\}$ are the *shape bases* and $\{c_{fk} \in \mathbb{R}\}$ are the *shape coefficients*.

Under this deformation model, the projection equation (1) can be rewritten as a projection equation from $\mathbb{P}^{3K}$ to $\mathbb{P}^2$ of the form

$$\lambda_{fp}\mathbf{x}_{fp} = \mathtt{M}_f \sum_{k=1}^{K}(c_{fk}\widehat{\mathbf{B}}_{kp}) + \mathbf{t}_f = \begin{bmatrix} c_{f1}\mathtt{M}_f \cdots \; c_{fK}\mathtt{M}_f \; \mathbf{t}_f \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{B}}_{1p} \\ \vdots \\ \widehat{\mathbf{B}}_{Kp} \\ 1 \end{bmatrix} = \mathtt{\Pi}_f\mathbf{B}_p. \tag{3}$$

Therefore, the matrix of image measurements $\mathtt{W}$ in (2) can be factorized into the product of a motion matrix $\mathtt{\Pi} \in \mathbb{R}^{3F\times(3K+1)}$ and a basis matrix $\mathtt{B} \in \mathbb{R}^{(3K+1)\times P}$ as

$$\mathtt{W} = \begin{bmatrix} \lambda_{11}\mathbf{x}_{11} \cdots \lambda_{1P}\mathbf{x}_{1P} \\ \vdots \quad\quad \vdots \\ \lambda_{F1}\mathbf{x}_{F1} \cdots \lambda_{FP}\mathbf{x}_{FP} \end{bmatrix} = \begin{bmatrix} c_{11}\mathtt{M}_1 \cdots c_{1K}\mathtt{M}_1 & \mathbf{t}_1 \\ \vdots \quad\quad\quad \vdots & \vdots \\ c_{F1}\mathtt{M}_F \cdots c_{FK}\mathtt{M}_F & \mathbf{t}_F \end{bmatrix} \begin{bmatrix} \widehat{\mathtt{B}}_1 \\ \vdots \\ \widehat{\mathtt{B}}_K \\ \mathbf{1}^\top \end{bmatrix} = \mathtt{\Pi B}\,. \tag{4}$$

Note that the motion matrix $\mathtt{\Pi}$ has the form $\mathtt{\Pi} = [\mathrm{diag}(\mathtt{M}_1, \ldots, \mathtt{M}_F)(\mathtt{C}\otimes \mathtt{I}_3) \mid \mathbf{t}]$, where $\mathbf{t} = \mathrm{stack}(\mathbf{t}_1, \ldots, \mathbf{t}_F)$. Furthermore, given the factorization in this form, we may read off the camera matrices $\mathtt{P}_f = [\mathtt{M}_f \mid \mathbf{t}_f]$ and the 3-D points from

$$\mathrm{stack}(\widehat{\mathtt{X}}_1, \ldots, \widehat{\mathtt{X}}_F) = (\mathtt{C} \otimes \mathtt{I}_3)\mathrm{stack}(\widehat{\mathtt{B}}_1, \ldots, \widehat{\mathtt{B}}_K)\,. \tag{5}$$

Note here, however, a basic ambiguity: the individual projection matrices can be determined from Π only up to independent scale factors, since scaling M_f can be balanced by a corresponding inverse scaling to the corresponding row of the coefficient matrix C.

Iterative methods. The rank constraint implied by (4) has been the basis for existing projective NRSM algorithms. As shown in [12], when the depths are known, the shape coefficients and shape basis may be computed from the factorization of W using a factorization technique similar to that in [8] for affine cameras. In [12], they solve the perspective reconstruction problem by alternately solving for the depths and the shape and motion parameters, in a similar way to [17]. In this paper, we seek an alternative purely algebraic solution to the problem that does not rely on any iterative optimization. In doing so, we are able to determine exactly what it is possible to compute uniquely, and what are the unavoidable ambiguities.

3 Nonrigid Shape and Motion Recovery

In this section, we propose a closed form solution to the NRSM problem from multiple perspective views. The key to our approach is to observe from equation (3) that the NRSM problem is a particular case of a reconstruction problem from $\mathbb{P}^{3K}$ to $\mathbb{P}^2$. This interpretation will allow us to solve directly for the motion matrix Π in (4) up to a projective transformation in $\mathbb{P}^{3K\times 3K}$, as we will show in §3.1. We will then propose an extremely simple linear algorithm for recovering the unknown projective transformation, hence the original camera matrices in $\mathbb{P}^{3\times 2}$, shape bases, and shape coefficients.

3.1 Recovery of the Projection Matrices $\mathbb{P}^{3K} \rightarrow \mathbb{P}^2$

While factorization methods such as [9,18,8] are commonly used in affine reconstruction problems involving affine or orthographic cameras, they are not so useful for reconstruction from perspective cameras, since they require iterative estimation of the depth values [11,17]. For such problems an alternative is to use tensor-based methods. The standard methods used for rigid structure and motion problems involve the fundamental matrix, trifocal or quadrifocal tensors [1]. It was shown in [14] that these tensor based methods can be extended to projections between projective spaces $\mathbb{P}^n$ and $\mathbb{P}^m$ of arbitrary dimensions with $n > m$. We will rely heavily on this method. In the particular case of relevance to the current problem, $n = 3K$ and $m = 2$.

In brief, given a suitable number of projections $\mathbb{P}^n \rightarrow \mathbb{P}^m$, we may compute a tensor that relates the coordinates of matching image points $\mathbf{x}_{fp}$ in $\mathbb{P}^m$. This tensor may be computed linearly, and from it the set of projection matrices Π_f may be extracted using non-iterative techniques. Subsequently, points $\mathbf{B}_p$ in $\mathbb{P}^n$ may be computed by triangulation such that $\lambda_{fp}\mathbf{x}_{fp} = \Pi_f\mathbf{B}_p$. Here, points $\mathbf{B}_p$ and the corresponding image points $\mathbf{x}_{fp}$ are expressed in homogeneous coordinates and the λ_{fp} are unknown scale factors, which do not need to be known for this reconstruction to be computed.

One may stack the projection matrices Π_f as well as the points $\mathbf{x}_{fp}$ on top of each other and form an equation

$$W = \text{stack}(\Pi_1, \ldots, \Pi_F)[\mathbf{B}_1 \ldots \mathbf{B}_P] = \Pi B\,, \tag{6}$$

which is of exactly the same form as the type of decomposition formulated in (4). It was shown in [14] that this factorization ΠB is unique except for the (non-significant) multiplication of each of the camera matrices Π_f by an arbitrary scale factor k_f and except for modifying ΠB to $\Pi A A^{-1} B$, where $A \in \mathbb{R}^{(3K+1)\times(3K+1)}$ is an invertible matrix. This is exactly analogous to the affine ambiguity inherent in affine factorization algorithms. However, here the matrix A represents a *projective* transformation, since we are using homogeneous coordinates. Thus, using tensors, we may achieve a similar factorization in the projective case as that computed by linear methods in the affine case. The only difference is that the number of views that may be used is restricted.

In the case of projective nonrigid motion, the image projection may be expressed as $\Pi_f : \mathbb{P}^{3K} \rightarrow \mathbb{P}^2$ and a factorization $W = \Pi B$ may be computed from any number of views between $(3K+1)/2$ and $3K+1$ (see [13]) using the tensor method. However, this does not produce a solution of the particular required form, given in (4) and it is impossible to extract the individual $\mathbb{P}^3 \rightarrow \mathbb{P}^2$ projection matrices immediately. We need to do some more work to find a matrix A that transforms each Π_f into the correct form. However, as will be seen, we gain from this since the remaining ambiguity is only affine or Euclidean (for calibrated cameras). Thus affine or Euclidean reconstruction is possible. How we enforce the correct form on the projection matrices Π_f will be the main focus of the rest of this paper.

3.2 Recovery of the Projective Transformation

As a result of our analysis in the previous subsection, at this point we have computed a projection matrix $\Pi \in \mathbb{R}^{3F\times(3K+1)}$. Our task is to transform this projection matrix by a matrix $A \in \mathbb{R}^{(3K+1)\times(3K+1)}$ such that ΠA is of the form $[\mathrm{diag}(M_1, \ldots, M_F)(C \otimes I_3) \mid \mathbf{t}]$ given in (4). To that end, we use the following steps.

Step 1. We assume that the matrix Π is full rank and, without loss of generality, that the top $3K \times 3K$ block of Π is non-singular. Hence, if we multiply Π by A_1, where $A_1^{-1} = [\Pi_{1:K}\ ;\ \mathbf{0}^\top, 1]$, we arrive at a matrix of a new form in which

$$(\Pi A_1)_{1:K} = \begin{bmatrix} I_K \otimes I_3 & \mathbf{0} \end{bmatrix}. \tag{7}$$

At this point, the first K row-blocks (in the block-representation) of ΠA_1 are of the desired form, but the remaining rows may be arbitrary.

Step 2. We multiply the matrix ΠA_1 by the block-diagonal matrix A_2, given by $A_2 = \mathrm{diag}(M_{K+1,1}, \ldots, M_{K+1,K}, 1)^{-1}$. Here the matrices $M_{K+1,k}$, $k = 1, \ldots, K$, are obtained from the $(K+1)$-st row-block of ΠA_1. Under a suitable assumption of genericity, these matrices will be non-singular, as will be seen in the proof of Theorem 1 below. This results in a matrix such that

$$(\Pi A_1 A_2)_{1:K+1} = \mathrm{diag}(M_1, \ldots, M_K, I_3) \begin{bmatrix} I_K \otimes I_3 & \mathbf{0} \\ I_3 \cdots I_3 & \mathbf{t}_{K+1} \end{bmatrix}, \tag{8}$$

where now the first $K+1$ row-blocks of $\Pi A_1 A_2$ are in the desired form and the $(K+1)$-st row-block contains only identity matrices.

Step 3. We are left with enforcing that the remaining $F-K-1$ row-blocks of $\Pi A_1 A_2$ have the desired algebraic structure by multiplying by a further matrix A_3. In order to

preserve the block diagonal structure of the top $3K \times 3K$ block of $\Pi A_1 A_2$, we can only multiply by a matrix A_3 whose top $3K \times 3K$ is also block diagonal. Therefore, we seek a matrix $A_3 = [\text{diag}(N_1, \cdots, N_K), \mathbf{0}\ ;\ {\mathbf{s}_1}^\top \cdots {\mathbf{s}_K}^\top, 1]$. In order for the $(K+1)$-st row-block to remain as identity matrices, it is easily verified that $N_k = I_3 - \mathbf{t}_{K+1}{\mathbf{s}_k}^\top$, so we need only compute the values of each $\mathbf{s}_k$.

For some $f > K+1$, let M_{fk} be the matrix in position (f,k) of $\Pi A_1 A_2$, and $\mathbf{t}_f$ be the vector in position $(f, K+1)$. By multiplication by A_3, M_{fk} is transformed to $M'_{fk} = M_{fk}(I_3 - \mathbf{t}_{K+1}{\mathbf{s}_k}^\top) + \mathbf{t}_f{\mathbf{s}_k}^\top$, which we may write as $M_{fk} + \mathbf{v}_{fk}{\mathbf{s}_k}^\top$, where the only unknown is $\mathbf{s}_k$. Our requirement on the form of the resulting matrix $\Pi A_1 A_2 A_3$ is that for each $f > K+1$ and $k > 1$ we have $c_{fk}^{-1} M'_{fk} = c_{f1}^{-1} M'_{f1}$ for some coefficients c_{fk}. This leads to equations

$$c_{fk}^{-1}(M_{fk} + \mathbf{v}_{fk}{\mathbf{s}_k}^\top) = c_{f1}^{-1}(M_{f1} + \mathbf{v}_{f1}{\mathbf{s}_1}^\top) \tag{9}$$

in which the unknowns are the vectors $\mathbf{s}_1, \ldots, \mathbf{s}_K$ and the coefficients c_{fk}^{-1}. Note that these equations are not linear. However, they may be written in the form

$$\frac{c_{f1}}{c_{fk}}(\mathbf{m}_{fk} + V_{fk}\mathbf{s}_k) = \mathbf{m}_{f1} + V_{f1}\mathbf{s}_1 \tag{10}$$

for suitable known matrices $V_{f1}, V_{fk} \in \mathbb{R}^{9\times 3}$ and vectors $\mathbf{m}_{f1}, \mathbf{m}_{fk} \in \mathbb{R}^9$. Multiplying this equation by a matrix $\Gamma_{fk} \in \mathbb{R}^{5\times 9}$ such that $\Gamma_{fk}\mathbf{m}_{fk} = 0$ and $\Gamma_{fk}V_{fk} = 0$ leads to $5(F-K)K$ linear equations in $\mathbf{s}_1$ of the form $\Gamma_{fk}V_{f1}\mathbf{s}_1 = -\Gamma_{fk}\mathbf{m}_{f1}$. Once $\mathbf{s}_1$ is known, one may rearrange (9) so that the equations become linear in the remaining $\mathbf{s}_k$ and coefficients c_{fk}/c_{f1}. Notice that there are many alternative ways of solving the equations in (9). Experimentation showed the the current method performs on par with other techniques.

3.3 Recovery of the Camera Matrices and of the Nonrigid Shape

After applying the transformation $A_1A_2A_3$ to Π, we obtain a matrix that is nominally of the desired form $\Pi' = [\text{diag}(M_1, \ldots M_F)(C \otimes I_3) \mid \mathbf{t}]$. Indeed, the first $K+1$ row blocks will be exactly of the desired form. However, because of measurement noise, the remaining blocks, corresponding to projections $\Pi'_f,\ f = K+2, \ldots, F$, will not be, so we need to correct this.

Consider a fixed frame $f > K+1$, and let $\Pi'_f = [M_{f1}, \ldots, M_{fK} \mid \mathbf{t}_f]$. This matrix will be nominally of the form $[c_{f1}M_f, \ldots, c_{fK}M_f \mid \mathbf{t}_f]$, but will be corrupted by noise. Each correspondence $M_{fk} = c_{fk}M_f$ may be seen as a set of 9 bilinear equations in the variables c_{fk} and M_f. We arrange the entries of all the M_{fk} into a matrix $E_{9\times K}$, one column for the entries of each M_{fk}. The set of all equations (for a fixed f) may then be written as $E_{9\times K} = \mathbf{m}_f{\mathbf{c}_f}^\top$ where ${\mathbf{c}_f}^\top = (c_{f1}, \ldots, c_{fK})$ and $\mathbf{m}_f$ is the vector of entries of the matrix M_f. We can then solve for $\mathbf{m}_f$ and $\mathbf{c}_f$ by computing the best rank-1 approximation of $E_{9\times K}$. Vectors $\mathbf{m}_f$ and $\mathbf{c}_f$ are computed up to a reciprocal scale ambiguity, which is all that is possible, as remarked previously.

By this method, we compute all M_f for $f > K+1$ and the corresponding shape coefficients c_{fk}. The resulting matrix Π'' will be exactly in the required true form. A solution for the shape bases B and projective depths λ_{fp} is then obtained by linear triangulation using equation (4). Finally, the nonrigid shape is given by $\hat{X} = (C \otimes I_3)\hat{B}$, and the camera matrices are $P_f = [M_f \mid \mathbf{t}_f]$.

4 Algorithm Justification

In the previous section, a method was given for transforming the matrix Π to the required form given in (4). However, there is no justification given that the resulting camera matrices and nonrigid shape will correspond to the ground truth. For instance, we did not show that the equations in (9) have a unique solution for the vectors $\mathbf{s}_k$, hence the matrix $\mathtt{A}_3$ may not be unique. In this section we show that, under suitable assumptions, the resulting product $\Pi\mathtt{A}_1\mathtt{A}_2\mathtt{A}_3$ is unique.

To that end, we make various definitions. A matrix $\Pi = \mathrm{stack}(\Pi_1, \ldots, \Pi_F)$ is said to be in *true form* if it is of the form $[\mathrm{diag}(\mathtt{M}_1, \ldots, \mathtt{M}_K)(\mathtt{C} \otimes \mathtt{I}_3) \mid \mathbf{t}]$, where all matrices $\mathtt{M}_f$ are invertible. A matrix is said to be in *canonical form* if it is of the form given in (8) with $\mathbf{t}_{K+1} \neq \mathbf{0}$, and in *true-canonical form* if it satisfies both conditions. We now state an important result.

Theorem 1. *Let $\Pi = \mathrm{stack}(\Pi_1, \ldots, \Pi_F)$ be a motion matrix, and assume that there exists $\mathtt{A}$ such that $\Pi\mathtt{A}$ is in true form. Subject to possible reordering of the rows Π_f of Π and under suitable assumptions of genericity, there exists a matrix $\mathtt{A}'$ such that $\Pi\mathtt{A}'$ is in true-canonical form. Furthermore, the true-canonical form is unique (for a fixed ordering of the rows Π_f).*

The meaning of the assumption of genericity will be made clear in the proof. Broadly speaking, it means that the motion of the camera is sufficiently general and independent of the shape deformation, and that the shape space is indeed K-dimensional, spanned by the K shape bases. In addition we assume that we can find $K+1$ frames such that no K of the corresponding shape matrices $\widehat{\mathtt{X}}_f$ are linearly dependent. We will order the frames so that these $K+1$ frames are numbered $1, \ldots, K+1$. The first K shapes will serve as the K shape bases.

Granted the truth of this theorem, the algorithm in the previous section will lead to the correct and unique solution. In particular, the matrix $\mathtt{A}_3$ used in step 3 must lead to a solution in true-canonical form. Therefore, the set of linear equations solved will have a unique solution.

The proof of Theorem 1 given here is of necessity brief. In a possible expanded version of this paper we can give more details, and in particular an exact analysis of the required genericity conditions.

Existence. For the existence part of the proof, it is clear that it is enough to show the existence of a matrix $\mathtt{A}'$ that transforms a true form matrix to one in true-canonical form. The steps of the proof follow the steps 1–2 of the algorithm of §3.2, except that we start with a matrix of the form $\Pi = [\mathrm{diag}(\mathtt{M}_1, \ldots, \mathtt{M}_F)(\mathtt{C} \otimes \mathtt{I}_3) \mid \mathbf{t}]$.

In the first step, the required transformation matrix will be of the form $[\,\mathtt{C}_{1:K} \otimes \mathtt{I}_3, \mathbf{t}_{1:K}\,;\ \mathbf{0}^\top, 1\,]^{-1}$. This will exist as long as $\mathtt{C}_{1:K}$ is invertible, which is the generic case, meaning that the shape matrices $\{\widehat{\mathtt{X}}_f \mid f = 1, \ldots, K\}$ span the complete shape space. If not, then we can reorder the frames so that this is so.

After the first step, the matrix remains in true form. Therefore, row $K+1$ will be of the form $\Pi_{K+1} = [c_1\mathtt{M}, \cdots, c_K\mathtt{M}, \mathbf{t}_{K+1}]$. We require that $\mathbf{t}_{K+1} \neq 0$, otherwise we rearrange the matrices Π_f so that this is so. The transformation matrix $\mathtt{A}_2 = \mathrm{diag}(c_1\mathtt{M}, \cdots, c_K\mathtt{M}, 1)^{-1}$ will transfer the matrix into canonical form. Observe that $\mathtt{M}$ is

invertible by our assumption that the matrix is in true form. If one of the c_k is zero, this means that the shape $\widehat{\mathtt{X}}_{K+1}$ at frame $K+1$ is in a space spanned by a proper subset of the shape bases $\widehat{\mathtt{B}}_k$, which we rule out by an appeal to genericity. Since we started with a matrix in true form, after these two steps, the matrix is now in true-canonical form. This completes the existence part of the proof.

Uniqueness. For the uniqueness part of the proof, consider a possible transformation $\mathtt{A}_3$, which transforms a matrix Π in true-canonical form to $\Pi' = \Pi\mathtt{A}_3$, also in true-canonical form. By the same argument as in §3.2, the matrix $\mathtt{A}_3$ must be of the form $\mathtt{A}_3 = [\mathrm{diag}(\mathtt{N}_1, \cdots, \mathtt{N}_K), \mathbf{0}\ ;\ \mathbf{s}_1{}^\top \cdots \mathbf{s}_K{}^\top, \lambda]$, with $\lambda \neq 0$ and $\mathtt{N}_k = \mathtt{I}_3 - \mathbf{t}_{K+1}\mathbf{s}_k{}^\top$. Applying this transform to the f-th row-block $\Pi_f = [c_{f1}\mathtt{M}_f, \ldots, c_{fK}\mathtt{M}_f, \mathbf{t}_f]$ of Π, results in a new block with entries $\mathtt{M}'_{fk} = c_{fk}\mathtt{M}_f(\mathtt{I}_3 - \mathbf{t}_{K+1}\mathbf{s}_k{}^\top) + \mathbf{t}_f\mathbf{s}_k{}^\top$. Since this new row-block must be in true form, for any two indices $1 \leq j, k \leq K$, there must exist constants c'_{fk} and c'_{fj} such that $c'^{-1}_{fk}\mathtt{M}'_{fk} = c'^{-1}_{fj}\mathtt{M}'_{fj}$. This leads to

$$c'_{fj}\left(c_{fk}\mathtt{M}_f(\mathtt{I}_3 - \mathbf{t}_{K+1}\mathbf{s}_k{}^\top) + \mathbf{t}_f\mathbf{s}_k{}^\top\right) = c'_{fk}\left(c_{fj}\mathtt{M}_f(\mathtt{I}_3 - \mathbf{t}_{K+1}\mathbf{s}_j{}^\top) + \mathbf{t}_f\mathbf{s}_j{}^\top\right), \quad (11)$$

which may be rewritten as

$$(c'_{fj}c_{fk} - c_{fj}c'_{fk})\mathtt{M}_f = c'_{fk}(\mathbf{t}_f - c_{fj}\mathtt{M}_f\mathbf{t}_{K+1})\mathbf{s}_j{}^\top - c'_{fj}(\mathbf{t}_f - c_{fk}\mathtt{M}_f\mathbf{t}_{K+1})\mathbf{s}_k{}^\top\ . \quad (12)$$

Since $\mathtt{M}_f$ is a matrix of rank 3, and the two terms on the right are of rank 1, this is impossible, unless $c'_{fj}c_{fk} - c_{fj}c'_{fk} = 0$ and

$$c_{fj}(\mathbf{t}_f - c_{fk}\mathtt{M}_f\mathbf{t}_{K+1})\mathbf{s}_k{}^\top = c_{fk}(\mathbf{t}_f - c_{fj}\mathtt{M}_f\mathbf{t}_{K+1})\mathbf{s}_j{}^\top, \quad (13)$$

where we have used the fact that $c_{fj}/c_{fk} = c'_{fj}/c'_{fk}$ to replace c'_{fk} by c_{fk}. Since the factorization of a rank-1 matrix is unique up to scaling the two factors, this relationship means that one of the following conditions must be true.

1. $\mathbf{t}_{K+1} = \mathbf{0}$. However, this is ruled out by hypothesis.
2. $\mathbf{t}_f = \mathbf{0}$. If this is so for all $f > K+1$ this implies that the position of these cameras are dependent on the position of the first K cameras. This is not a generic camera motion.
3. The vectors $\mathbf{t}_f$ and $\mathtt{M}_f\mathbf{t}_{K+1}$ are linearly dependent. However, if this is true for all $f > K+1$, then it implies that $\mathtt{M}_f^{-1}\mathbf{t}_f$ is a multiple of $\mathbf{t}_{K+1}$. This is a non-generic camera motion, since $-\mathtt{M}_f^{-1}\mathbf{t}_f$ is the position of the camera at frame f.
4. Finally, if $\mathtt{M}_f\mathbf{t}_{K+1}$ and $\mathbf{t}_f$ are linearly independent, then in order for the vectors $\mathbf{t}_f - c_{fk}\mathtt{M}_f\mathbf{t}_{K+1}$ and $\mathbf{t}_f - c_{fj}\mathtt{M}_f\mathbf{t}_{K+1}$ to differ only by a scale factor, it is necessary that $c_{fj} = c_{fk}$. This means that the f-th row-block of Π must be of the form $\Pi_f = [c_f\mathtt{I}_3, c_f\mathtt{I}_3, \ldots, c_f\mathtt{I}_3, \mathbf{t}_f]$, with all the coefficients c_{fk} the same along this row. Apart from a constant scale, these are the same set of coefficients as for the $(K+1)$-st row-block, which means that the shape of the scene is the same for this frame as for frame $K+1$. If this is true for all $f > K+1$, it implies that the object has the same shape, and does not deform for all of the frames $K+1$ to F.

If on the other hand, the deformation of the scene is generic, then none of the conditions given above can be fulfilled. In this case the canonical form is uniquely determined. This concludes the proof.

5 Affine and Euclidean Shape Reconstruction

Having established the correctness of the proposed reconstruction algorithm, we now turn to the question of uniqueness of the reconstructed shapes. In particular, we show that, even though there are ambiguities in the reconstruction of shape bases and shape coefficients, the reconstructed shape is actually unique. Moreover, we will show that when $K \geq 2$, one recover the shape up to an affine transformation, which represents a significant improvement with respect to the case $K = 1$, where one can only recover the shape up to a projective transformation.

Affine Shape Reconstruction. As a consequence of the proof of the uniqueness result of Theorem 1, if Π and Π' are two matrices in true form, then they are related by $\Pi' = \Pi A$, where A is some product of matrices of the form

$$A_1 = \begin{bmatrix} C' \otimes I_3 & 0 \\ \mathbf{0}^\top & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} I_3 \otimes I_3 & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \quad A_3 = \begin{bmatrix} I_3 \otimes M & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} \tag{14}$$

(not the same as the matrices A_1, A_2, A_3 in §3.2) and their inverses. Here C' has dimension a $K \times K$, and the product matrix A may be written as $A = [C' \otimes M, \mathbf{t}\,;\, \mathbf{0}^\top, 1]$.

In the factorization of $W = \Pi B$, the inverse transformations are applied to B. The first of these transformations causes a change of the shape bases through linear combinations. However, it does not change the shape of the points $\mathbf{X}_{fp}$. To see this, observe that the corresponding change to $\widehat{B}$ is to replace it by $(C' \otimes I_3)^{-1}\widehat{B}$. At the same time, the coefficients in the representation (4) of Π are multiplied by C'. However, from (5) X is unchanged by this operation, since $\widehat{X} = (C \otimes I_3)\widehat{B} = (C \otimes I_3)(C' \otimes I_3)(C' \otimes I_3)^{-1}\widehat{B}$ so the matrix $(C' \otimes I_3)$ cancels with its inverse, leaving $\widehat{X}$ unchanged.

The other two transformations effect an affine transformation of the shape bases. By an application of the transformation A_2 each of the shape bases may be translated so that the points it consists of have their centroid at the origin. The resulting reconstruction will be called "centred". Since each of the shape bases is centred at the origin, so will the sets of points X_f at any other frame, since they are linear combinations of the shape bases. If desired, the reconstruction ΠB may be centred by applying a transformation $\Pi \rightarrow \Pi A_4$ and $B \rightarrow A_4^{-1}B$, where $A_4^{-1} = [I_K \otimes I_3, -\mathbf{w}\,;\, \mathbf{0}^\top, 1]$, and $\hat{\mathbf{w}} = \mathrm{stack}(\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_K)$ is made up of the centroids $\hat{\mathbf{w}}_k$ of the points $\widehat{\mathbf{B}}_{k1}, \ldots, \widehat{\mathbf{B}}_{kP}$ in each shape basis $\widehat{B}_k$. A centred reconstruction is unique up to a common linear transformation of all of the shape bases and a corresponding transformation of each of the camera matrices.

We see that the reconstruction is unique except for the following ambiguities.

1. Individual scaling applied to each frame independently, as pointed out in §2.
2. Individual translations of each of the shape bases. Thus, there is a K-fold translation ambiguity in the global reconstruction over all frames. This ambiguity may be removed by computing a centred reconstruction, or by assuming that the first K translations are zero. However, observe that the obtained translations do not necessarily correspond to the ground truth.
3. An overall linear transformation. In the case of calibrated cameras, this is an overall global rotation with respect to a global coordinate frame, which of course can not be determined.

Euclidean Shape Reconstruction. If the cameras are calibrated, we may assume that they are of the form $\mathtt{P}_i = [\mathtt{R}_i \mid -\mathtt{R}_i\mathbf{t}_i]$, where each of the $\mathtt{R}_i$ is a rotation. In this case, any initially computed motion matrix Π will be equivalent (under multiplication by $\mathtt{A}$) to a *Euclidean true form* motion matrix (4), meaning all the $\mathtt{M}_f$ are rotations. Furthermore, the details of the existence part of Theorem 1 show that Π is then equivalent to a Euclidean true-canonical form matrix. Since the true-canonical form is unique, this shows that Euclidean reconstruction is possible and unique. Furthermore, the algorithm of §3 will naturally lead virtually without modification to the correct Euclidean solution. The details are simple to verify.

Autocalibration. It is interesting and somewhat surprising that for $K \geq 2$ our algorithm gives an affine reconstruction even from uncalibrated cameras. This contrasts with the rigid motion case ($K = 1$), where the reconstruction is only projective. It is easily seen that the affine reconstruction is easily upgraded to a Euclidean reconstruction using standard linear autocalibration techniques. Indeed in the standard method of stratified reconstruction and autocalibration the upgrade from projective to affine reconstruction is difficult, but to upgrade from affine to Euclidean, given mild assumptions on common parameters of the cameras is simple and linear. Details may be found in [1].

6 Experiments

Synthetic Data. We first evaluate our algorithm on synthetically generated data. The $K = 2, 3$ shape bases are generated by randomly drawing P 2-D points uniformly on $[-1, 1]\times[-1, 1]$ and then scaling these points with a depth uniformly drawn in the range of 100-400 units of focal length (u.f.l.). The shape coefficients are also randomly drawn from a uniform distribution in $[-1, 1]$. The 3-D points are then generated by taking a linear combination of the shape bases with the shape coefficients. These points are rotated and translated according to rigid-body motions with a random axis of rotation and a random direction of translation. $F = 4$ to 6 perspective views are obtained by projecting these points onto an image with 1000×1000 pixels. Zero-mean Gaussian noise with a standard deviation of $\sigma \in [0, 2]$ pixels is added to the so-obtained point correspondences.

We evaluate the accuracy of our algorithm with respect to four factors: amount of noise, number of shape bases, number of frames, and number of point correspondences. The performance measures are the angles between the estimated rotations and translations $(\hat{\mathtt{M}}_f, \hat{\mathbf{t}}_f)$ and the ground truth $(\mathtt{M}_f, \mathbf{t}_f)$,

$$\theta_{\mathtt{M}} = \frac{1}{F}\sum_{f=1}^{F} \arccos\big((\mathrm{trace}(\hat{\mathtt{M}}_f^\top \mathtt{M}_f) - 1)/2\big),\ \theta_{\mathbf{t}} = \frac{1}{F-K}\sum_{f=K+1}^{F} \arccos\big(\hat{\mathbf{t}}_f^\top \mathbf{t}_f\big),$$

averaged over 1000 trials. Note that, due to the reconstruction ambiguities, we assume that the first K translations are zero. Moreover, recall that the remaining translations are computed up to one scale factor per frame, hence the choice of the angle between the true and estimated translation as an error measure.

Figure 1 shows average error versus amount of noise plots for several choices of the parameters. The number of points is chosen either as $P = 200$, or as twice the

minimum number of points needed to reconstruct the multifocal tensor, i.e., $P \geq 3^F / \prod_{f=1}^{F}(3 - \pi_f)$, where $\pi_f \in \{1, 2\}$ defines the tensor profile for the f-th frame. As expected, the error increases with the amount of noise and reduces with the number of points correspondences. However, the error does not necessarily reduce as the number of frames increases. When $K = 2$, this can be seen by comparing the curves for $(F, P) = (4, 200)$ and $(F, P) = (4, 82)$, with those for $(F, P) = (5, 200)$ and $(F, P) = (5, 62)$, respectively. This is because the number of unknowns in the multifocal tensor increases exponentially with the number of frames, and a number of nonlinear constraints on the entries of the tensor are neglected when computing and factorizing this tensor using linear techniques. Notice also by comparing the curves for $(K, F, P) = (2, 5, 62)$ and $(K, F, P) = (3, 5, 486)$ that the error reduces as the number of shape bases increases. However, the improvement comes at the cost of increasing the number of points needed. Indeed, when the number of points is increased from 62 to 200, the performances for $(K, F) = (2, 5)$ and $(K, F) = (3, 5)$ are comparable. Finally, notice also that the best existing affine algorithm by Xiao et al. [8] does not perform well on perspective data. This algorithm requires a minimum of $F \geq K^2 + K$ images, so we only evaluate it for $(K, F) = (2, 6)$. Our algorithm, on the other hand, requires a minimum number of frames of $F \geq (3K + 1)/2$.

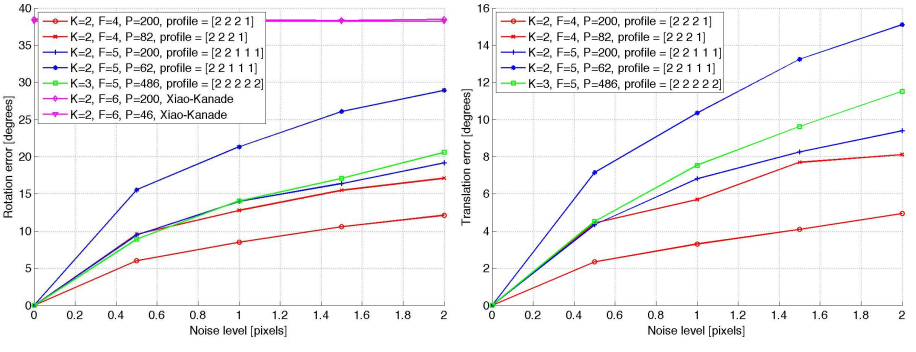


Fig. 1. Reconstruction errors as a function of noise, number of shape basis, number of frames, and number of point correspondences

Real Data. We now test the performance of our algorithm on a video sequence containing two hands moving in front of a static background shown in Fig. 2. The sequence is taken from [13], and consists of $F = 5$ views taken by a moving camera observing 8 points on the static background and another 32 points on the gesturing hands. The 8-point algorithm was used to compute the ground truth camera motion from the 8 static points. We then applied our algorithm and the algebraic algorithm of [13] for $K = 2$ shape basis and $F = 5$ views. We chose the first image as the reference. The errors in the estimation of the rotations are shown in Table 6. Note that our algorithm outperforms that in [13] for 3 out of 4 frames. Translation errors are not computed, as with real sequences one cannot assume zero translations for the first K frames.

Fig. 2. Frames 1-5 of a sequence of gesturing hands used in [13]

Table 1. Errors in the estimation of the rotations for a sequence of gesturing hands

Frame	2	3	4	5
Quintifocal method [13]	0.1644°	5.9415°	2.5508°	54.5860°
Our method	5.5174°	0.6773°	0.1642°	27.1583°

7 Discussion and Conclusions

We have presented several theoretical results pertaining to the nonrigid shape and motion problem from multiple perspective views. Most notably, we have shown that a highly multilinear problem admits a closed form, linear solution. Furthermore, we highlighted several similarities and differences between the rigid and nonrigid case.

While our theoretical framework does provide an algorithm for solving the reconstruction problem, we did not explore algorithmic aspects in this paper, such as robustness to noise or outliers. The reader can see that our proposed method is very simple, involving essentially a series of matrix multiplications. Each one of those steps can be made robust. We argue that the real bottleneck with the current method is not in our approach, but rather in the tensor estimation and factorization approach of [14]. Improving on the robustness of these methods is an interesting avenue for future research.

Acknowledgments. Richard Hartley has been supported by NICTA, which is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program. René Vidal has been supported by startup funds from JHU, by grants NSF CAREER IIS-0447739, NSF EHS-0509101, and ONR N00014-05-10836, and by contract JHU APL-934652.

References

1. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge (2004)
2. Ma, Y., Soatto, S., Kosecka, J., Sastry, S.: An Invitation to 3D Vision: From Images to Geometric Models. Springer, Heidelberg (2003)
3. Brand, M.: Morphable 3D models from video. In: Conference on Computer Vision and Pattern Recognition, pp. 456–463 (2001)
4. Brand, M., Bhotika, R.: Flexible flow for 3D nonrigid tracking and shape recovery. In: Conference on Computer Vision and Pattern Recognition, pp. 315–322 (2001)
5. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3D shape from image streams. In: Conference on Computer Vision and Pattern Recognition, pp. 2690–2696 (2000)

6. Torresani, L., Bregler, C.: Space-time tracking. In: European Conference on Computer Vision, pp. 801–812 (2002)
7. Torresani, L., Yang, D., Alexander, E., Bregler, C.: Tracking and modeling non-rigid objects with rank constraints. In: Conference on Computer Vision and Pattern Recognition, pp. 493–500 (2001)
8. Xiao, J., Chai, J., Kanade, T.: A closed-form solution to non-rigid shape and motion recovery. International Journal of Computer Vision 67, 233–246 (2006)
9. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography. International Journal of Computer Vision 9, 137–154 (1992)
10. Xiao, J., Kanade, T.: Non-rigid shape and motion recovery: Degenerate deformations. In: Conference on Computer Vision and Pattern Recognition, pp. 668–675 (2004)
11. Sturm, P., Triggs, B.: A factorization based algorithm for multi–image projective structure and motion. In: European Conference on Computer Vision, pp. 709–720 (1996)
12. Xiao, J., Kanade, T.: Uncalibrated perspective reconstruction of deformable structures. In: IEEE International Conference on Computer Vision, pp. 1075–1082 (2005)
13. Vidal, R., Abretske, D.: Nonrigid shape and motion from multiple perspective views. In: European Conference on Computer Vision, pp. 205–218 (2006)
14. Hartley, R., Schaffalitzky, F.: Reconstruction from projections using Grassmann tensors. In: European Conference on Computer Vision, pp. 363–375 (2004)
15. Oliensis, J., Hartley, R.: Iterative extensions of the Sturm/Triggs algorithm: convergence and nonconvergence. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 2217–2233 (2007)
16. Aanaes, H., Kahl, F.: Estimation of deformable structure and motion. In: ECCV Workshop on Vision and Modelling of Dynamic Scenes (2002)
17. Mahamud, S., Hebert, M., Omori, Y., Ponce, J.: Provably-convergent iterative methods for projective structure from motion. In: Conference on Computer Vision and Pattern Recognition, vol. I, pp. 1018–1025 (2001)
18. Costeira, J., Kanade, T.: A multibody factorization method for independently moving objects. International Journal of Computer Vision 29, 159–179 (1998)

Shadows in Three-Source Photometric Stereo

Carlos Hernández[1], George Vogiatzis[1], and Roberto Cipolla[2]

[1] Computer Vision Group, Toshiba Research Europe, Cambridge, UK
[2] Dept. of Engineering, University of Cambridge, Cambridge, UK

Abstract. Shadows are one of the most significant difficulties of the photometric stereo method. When four or more images are available, local surface orientation is overdetermined and the shadowed pixels can be discarded. In this paper we look at the challenging case when only three images under three different illuminations are available. In this case, when one of the three pixel intensity constraints is missing due to shadow, a 1 dof ambiguity per pixel arises. We show that using integrability one can resolve this ambiguity and use the remaining two constraints to reconstruct the geometry in the shadow regions. As the problem becomes ill-posed in the presence of noise, we describe a regularization scheme that improves the numerical performance of the algorithm while preserving data. We propose a simple MRF optimization scheme to identify and segment shadow regions in the image. Finally the paper describes how this theory applies in the framework of color photometric stereo where one is restricted to only three images. Experiments on synthetic and real image sequences are presented.

1 Introduction

Photometric stereo is a well established 3d reconstruction technique based on the powerful shading cue. A sequence of images (typically three or more) of a 3d scene are obtained from the same viewpoint and under varying illumination. From the intensity variation in each pixel one can estimate the local orientation of the surface that projects onto that pixel. By integrating all these surface orientations a very detailed estimate of the surface geometry can be obtained. As any other reconstruction method, photometric stereo faces several difficulties when faced with real images. One of the most important of these difficulties is the frequent presence of shadows in an image. No matter how careful the arrangement of the light sources, shadows are an almost unavoidable phenomenon, especially in objects with complex geometries. This paper investigates the phenomenon of shadows in photometric stereo with three light sources.

Shadows in photometric stereo have been the topic of a number of papers [1,2,3]. Most papers assume we are given four or more images under four different illuminations. This over-determines the local surface orientation and albedo (3 degrees of freedom) which implies that we can use the residual of some least squares solution, to determine whether shadowing has occurred. However when we are only given three images there are no *spare* constraints against which to

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 290–303, 2008.

test our hypothesis. Therefore the problem of detecting shadows becomes more difficult. Furthermore, when a pixel is in shadow in one of the three images most methods simply discard it. We show how one can use the remaining two image intensity measurements to estimate the surface geometry inside the shadow region. The solution we propose is based on enforcing (1) integrability of the gradient field, as well as (2) smoothness in the recovered intensity of the missing channel.

Using photometric stereo on just three images may seem like an unreasonably hard restriction. There is however a particular situation when only three images are available. This technique is known as color photometric stereo [4] and it uses three light sources with different light spectra. When the scene is photographed with a color camera, the three color channels capture three different photometric stereo images. Because shape acquisition is performed on each frame independently, the method can be used on video sequences without having to change illumination between frames [5]. In this way we can capture the 3D shape of deforming objects such as cloth, or human faces. Since the method is constrained to operate only on three images it is an ideal application of the theory we present here. Summarizing, the two main contributions of this paper are the following:

- We show how to exploit image regions in photometric stereo where one of the three images is in shadow. A geometric formulation of the problem is given where a set of point-to-point and point-to-line distances are minimized under the integrability condition.
- We develop a regularization scheme that makes the optimization problem well posed while not suppressing the data. This scheme is successfully validated within the color photometric stereo technique of [5].

1.1 Previous Work

A vast literature exists on the topic of photometric stereo. Its applications range from 3D reconstruction [6], medical imaging [7] or cloth modeling [5]. One of the main limitations of photometric stereo is the number of different lights required and how the algorithm copes with highlights or shadows.

A minimum of 3 lights is required to perform photometric stereo with no extra assumptions [6], and only 2 lights with the additional assumption of constant albedo [8]. Whenever more lights are available, the light visibility problem becomes a labeling problem where each point on the surface has to be assigned to the correct set of lights in order to successfully reconstruct the surface.

For objects with constant albedo, [3] used a Rank-2 constraint to detect surfaces illuminated by only 2 lights. In the case of general albedo, every point on the surface has to be visible in at least 3 images. A 4-light photometric stereo setup was proposed in [9], where light occlusion was detected by checking the consistency of all the possible triplets of lights. The work by [10] was able to detect light occlusions in a 4-light setup and simply treat them as outliers. In [1] a similar algorithm to [9] is presented using a 4-light colored photometric stereo approach.

In the recent work by [2], an iterative MRF formulation is proposed for detecting light occlusion and exploiting it as a surface integration constraint. However, the algorithm also requires a minimum of 4 lights and is targeted for setups with a large number of lights.

In this paper we propose a novel solution for 3-light photometric stereo with shadows and varying albedo. We are able to detect and exploit photometric stereo constraints with only two lights while the constant albedo constraint is relaxed into a more practical smoothly varying albedo constraint.

2 Three-Source Photometric Stereo with Shadows

In classic three-source photometric stereo we are given three images of a scene, taken from the same viewpoint, and illuminated by three distant light sources. The light sources emit the same light frequency spectrum from three different non-coplanar directions. We will assume an orthographic camera (with infinite focal length) for simplicity, even though the extension to the more realistic projective case is straightforward [11]. In the case of orthographic projection one can align the world coordinate system so that the xy plane coincides with the image plane while the z axis corresponds to the viewing direction. The surface in front of the camera can then be parameterized as a height function $Z(x, y)$. If Z_x and Z_y are the two partial derivatives of Z one can define the vector

$$\mathbf{n} = \frac{1}{\sqrt{Z_x^2 + Z_y^2 + 1}} \begin{pmatrix} Z_x & Z_y & -1 \end{pmatrix}^\top$$

that is locally normal to the surface at (x, y). For $i = 1 \ldots 3$ let $c_i(x, y)$ denote the pixel intensity of pixel (x, y) in the i-th image. We assume that in the i-th image the surface point $\begin{pmatrix} x & y & Z(x,y) \end{pmatrix}^\top$ is illuminated by a distant light source whose direction is denoted by the vector $\mathbf{l}_i$ and whose spectral distribution is $E(\lambda)$. We also assume that the surface point absorbs incoming light of various wavelengths according to the reflectance function $R(x, y, \lambda)$. Finally, let the response of the camera sensor at each wavelength be given by $S(\lambda)$. Then the pixel intensity $c_i(x, y)$ is given by

$$c_i(x, y) = \left(\mathbf{l}_i^\top \mathbf{n}\right) \int E(\lambda)\, R(x, y, \lambda)\, S(\lambda)\, d\lambda. \tag{1}$$

The value of this integral is known as the surface *albedo* of point $\begin{pmatrix} x & y & Z(x,y) \end{pmatrix}^\top$. We can define the albedo-scaled normal vector

$$\mathbf{b} = \mathbf{n} \int E(\lambda)\, R(x, y, \lambda)\, S(\lambda)\, d\lambda$$

so that (1) becomes a simple dot product

$$c_i = \mathbf{l}_i^\top \mathbf{b}. \tag{2}$$

Photometric stereo methods use the linear constraints of (2) to solve for $\mathbf{b}$ in a least squares sense. From this they obtain the partial derivatives of the height

function which are integrated to produce the function itself. In three-source photometric stereo, when the point is not in shadow with respect to all three lights, we measure three positive intensities c_i each of which gives a constraint on $\mathbf{b}$. If we write $L = \left[\mathbf{l}_1\ \mathbf{l}_2\ \mathbf{l}_3\right]^\top$ and $\mathbf{c} = \left[c_1\ c_2\ c_3\right]^\top$ then the system has exactly one solution which is given by

$$L^{-1}\mathbf{c}. \tag{3}$$

If the point however is in shadow, say in the i-th image, then the measurement of c_i cannot be used as a constraint. Since each equation (2) describes a 3D plane, the intersection of the two remaining constraints is a 3D line. If $\mathbf{e}_i$ is the i-th column of the 3×3 identity matrix and D_i is the identity matrix with a zero in the i-th position in the diagonal, the possible solutions for $\mathbf{b}$ are

$$L^{-1}D_i\mathbf{c} + \mu L^{-1}\mathbf{e}_i. \tag{4}$$

where μ is a scalar parameter. This parameter represents the value that c_i would have, had the point not been in shadow in the i-th image.

3 Projection to 2D Space

At this point it is useful to project the scaled normals $\mathbf{b}_j$ into 2D space with coordinates p and q. We define the following operator: $\mathcal{P}[\mathbf{x}] = (x_1/x_3,\ x_2/x_3)$. Let the scaled normal $\mathbf{b}$ of surface point $(x, y, Z(x, y))$ project to point $\mathcal{P}[\mathbf{b}] = (p, q)$. Then the coordinates p and q are equal to the two partial derivatives Z_x and Z_y respectively. According to the image constraints and assuming no noise in the data, we can have one of the following three cases:

1. The surface point is in shadow in two or more images. In this case there is no constraint in $\mathcal{P}[\mathbf{b}]$ from the images.
2. The surface point is not in shadow in any of the three images. In this case (p, q) coincides with $\mathcal{P}\left[L^{-1}\mathbf{c}\right] = (P, Q)$.
3. The surface point is in shadow in exactly one image, say the i-th. In this case (p, q) must lie on the line that joins $\mathcal{P}\left[L^{-1}D_i\mathbf{c}\right] = \left(\overline{P}, \overline{Q}\right)$ and $P\left[L^{-1}\mathbf{e}_i\right] = \left(P_e^{(i)}, Q_e^{(i)}\right)$. We call this line the *shadow line* of the shaded pixel.

Note that for all pixels j that are occluded under the i-th image, the corresponding points $\left(\overline{P}_j, \overline{Q}_j\right)$ are on the line described by equation $\mathbf{l}_i^\top \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$. Also, the shadow lines of all those pixels intersect at the point $\left(P_e^{(i)}, Q_e^{(i)}\right)$.

Now in the presence of noise in the data $\mathbf{c}$, cases 2 and 3 above do not hold exactly as points (P, Q) and $\left(\overline{P}, \overline{Q}\right)$ are corrupted: The point (p, q) is slightly different from (P, Q) for unoccluded pixels, and $\left(\overline{P}, \overline{Q}\right)$ is not exactly on the line joining (p, q) and $\left(P_e^{(i)}, Q_e^{(i)}\right)$. Figure 1 shows the configuration for six pixels where pixel 1 is under shadow in the i-th image while pixel 2 is not in shadow in any image.

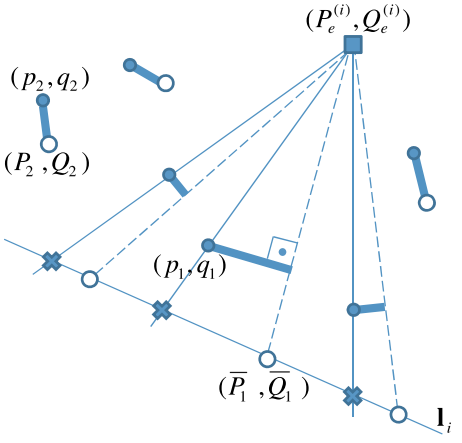


Fig. 1. Geometry of shadowed pixels. The points (p_j, q_j) (dark dots) represent the partial derivatives of the height function at pixel j. For each point (p_j, q_j) there is a corresponding data point (white dot). Pixel 2 is unoccluded and hence (p_2, q_2) must be as close as possible to its data point (P_2, Q_2). Pixel 1 however is occluded so (p_1, q_1) must be as close as possible to its *shadow line*. This is the line joining its data point $(\overline{P}_1, \overline{Q}_1)$ and the intersection point $\left(P_e^{(i)}, Q_e^{(i)}\right)$.

3.1 Integrability Condition

Note that the (p, q) coordinates are not independent for each pixel as they represent the partial derivatives of a scalar field and as a result they must satisfy an integrability condition. By assuming that the height function is a continuous piecewise polynomial, the integrability condition takes the form of a convolution between p and q. If for example $Z(x, y)$ is a linear interpolation between height values at pixel centers $Z(i, j)$ then we can express integrability as

$$p\,(i+1, j) - p\,(i, j) = q\,(i, j+1) - q\,(i, j)\,.$$

Our strategy is to obtain values (p, q) for each pixel that minimize the discrepancy between the noisy data and the model while satisfying the integrability condition. This is discussed in the following section.

4 Integrating in the Shadow Regions

As mentioned, under noise in the image data $\mathbf{c}$, the 2D points (P, Q) and $\left(\overline{P}, \overline{Q}\right)$ are not perfectly consistent with the model. For non-shadowed pixels, difference between model and data can be measured by the point-to-point square difference term

$$\mathcal{E} = (p - P)^2 + (q - Q)^2\,.$$

In the case of the shadowed pixels however we have a choice of possible ways to quantify the non-collinearity of (p, q), $\left(\overline{P}, \overline{Q}\right)$ and $\left(P_e^{(i)}, Q_e^{(i)}\right)$. Ideally, since

$\left(\overline{P},\overline{Q}\right)$ is the point that contains the noise, we should be measuring the distance from $\left(\overline{P},\overline{Q}\right)$ to the intersection of the line joining (p,q) and $\left(P_e^{(i)},Q_e^{(i)}\right)$ with the line $\mathbf{l}_i^\top \begin{pmatrix} p \\ q \\ 1 \end{pmatrix} = 0$. This leads to the following distance term

$$\left\| \left(\overline{P},\overline{Q}\right) - \mathcal{P}\left[\left(I - \mathbf{m}_i\mathbf{l}_i^\top\right)\mathbf{b}\right] \right\|^2$$

where $\mathbf{m}_i$ is the i-th vector of L^{-1}. The expression inside the square is non linear with respect to p and q. We therefore choose to minimize the distance from (p,q) to the line joining $\left(\overline{P},\overline{Q}\right)$ and $\left(P_e^{(i)},Q_e^{(i)}\right)$. This distance is shown in the red dotted line in figure 1. The term this corresponds to is

$$\mathcal{E}_o^{(i)} = \frac{\left(\left(\overline{Q} - Q_e^{(i)}\right)\left(p - P_e^{(i)}\right) - \left(\overline{P} - P_e^{(i)}\right)\left(q - Q_e^{(i)}\right)\right)^2}{\left(\overline{P} - P_e^{(i)}\right)^2 + \left(\overline{Q} - Q_e^{(i)}\right)^2}$$

and here the quantity being squared is linear with respect to p and q.

So we now assume we are given a labeling of pixels into all the possible types of shadow. Let $\mathcal{S}$ contain all non-shadowed pixels while $\mathcal{S}_i$ contains pixels shaded in the i-th image. Our cost function becomes

$$\sum_{j\in\mathcal{S}} \mathcal{E}_j + \sum_{j\in\mathcal{S}_1} \overline{\mathcal{E}}_j^{(1)} + \sum_{j\in\mathcal{S}_2} \overline{\mathcal{E}}_j^{(2)} + \sum_{j\in\mathcal{S}_3} \overline{\mathcal{E}}_j^{(3)}$$

which is a set of quadratic terms in p and q. Finding the minimum of this quantity subject to the integrability condition is a constrained linear least squares problem that can be solved by a variety of methods [12].

Figure 2 shows this idea applied in practice on synthetic data. This experiment indicates that the overall geometry seems to be reconstructed quite well in shadowed regions, provided that these are surrounded by unshaded pixels. The latter act as boundary conditions for the shadowed regions and give the problem a unique solution. Furthermore it also provides evidence that in its present form the problem is ill-conditioned, especially in larger shadowed regions. The following section sheds more light on this and describes our proposed remedy.

4.1 Regularization

The linear least squares optimization framework described in section 2 when executed in practice shows signs of ill-posedness in the presence of noise. This is demonstrated in the synthetic case of figure 2 where three images of a sphere have been generated. Three shadow regions corresponding to each of the three lights have been introduced. Even though the overall shape of the object is accurately captured some characteristic 'scratch' artifacts are observed. These are caused by the point-to-line distances which do not introduce enough constraints in the cost function. The point (p,q) can move significantly in a direction parallel to the corresponding shadow line only to gain a slight decrease in the overall cost. This

results in violent perturbations in the resulting height function that manifest themselves as deep scratches running perpendicular to the shadow lines. The solution to this is some type of regularization on the space of solutions. We have two main requirements on the choice of regularizing criterion:

- The scheme must be consistent with the linear least squares framework. No non-linear constraints can be enforced.
- It must suppress noise while preserving as much of the data as possible.

One possible choice for a regularization criterion is minimizing the Laplacian of the height field $\nabla^2 z$. This is known to have good noise reduction properties and to produce smooth well behaved surfaces with low curvature. However, the Laplacian is isotropic so it tends to indiscriminately smooth along all possible directions. See [13] for a good discussion of anisotropic alternatives to Laplacian filtering in the context of gradient field integration. In the case of our problem, we would like to enforce regularization in (p, q) space along the direction of the shadow line for each shadowed pixel. Fortunately there is an efficient way of achieving this that satisfies both our goals. For this we need to modify our point to line distance term to the following:

$$\hat{\mathcal{E}}^{(i)} = \left(p - w\overline{P} - (1-w)\, P_e^{(i)}\right)^2 + \left(q - w\overline{Q} - (1-w)\, Q_e^{(i)}\right)^2. \tag{5}$$

This introduces a new variable w per shaded pixel, that specifies a location along the shadow line of that pixel. The term is still quadratic with respect to p, q and w but this now allows us to regularize the solution in a meaningful way. The variable w is related to parameter μ of (4) by

$$w = \frac{\mathbf{e}_3^\top L^{-1} D_i \mathbf{c}}{\mathbf{e}_3^\top L^{-1} D_i \mathbf{c} + \mu \mathbf{e}_3^\top L^{-1} \mathbf{e}_i}. \tag{6}$$

As we mentioned, μ represents the value of c_i that would have been measured had the pixel not been in shadow in that image. We propose putting a cost on the length of ∇w inside the shaded regions. As w is a proxy for μ, this corresponds to introducing smoothness in $\mathbf{l}_i^\top \mathbf{b}$. We can therefore eliminate the scratch artifacts while letting $\mathbf{b}$ have variability in the directions perpendicular to $\mathbf{l}_i$. Figure 2 shows that this scheme works quite well in practice.

Throughout all of the previous discussion we have assumed knowledge of labeling of pixels according to shadows. The next section discusses how we propose to segment shadow regions in the image.

4.2 Segmenting Shadow Regions

It is known [1] that in photometric stereo with four or more images one can detect shadows by computing the scaled normal that satisfies the constraints in a least squares sense. If the residual of this least squares calculation is high, this implies that the pixel is either in a shadow or in a highlight. With three images however this becomes impossible as the three constraints can always be satisfied

exactly, leaving a residual of zero. Recently, [2] proposed a graph-cut based scheme for labeling shadows in photometric stereo with four or more images. Based on the constraint residual, they compute a cost for assigning a particular shadow label to each pixel. This cost is then regularized in an MRF framework where neighboring pixels are encouraged to have *similar* shadow labels. We would like to use a similar framework but we must supply a different cost for assigning a shadow label. The basic characteristic of a shadow region is that pixel intensities inside it are dark. However this can also occur because of dark surface albedo. To remove the albedo factor we propose to divide pixel intensities with the magnitude of the intensity vector $\mathbf{c}$. Our cost for deciding that a pixel is occluded in the i-th image is $c_i / \|\mathbf{c}\|$. This still leaves the possibility that we mistakenly classify a pixel whose normal is nearly perpendicular to the i-th illumination direction $\mathbf{l}_i$. However in that case the pixel is close to being in a self shadow so the risk from misclassifying it is small. The cost for assigning a pixel to the non-shadowed set is given by

$$\sqrt{3} - \min_i \frac{c_i}{\|\mathbf{c}\|}.$$

We regularize these costs in an MRF framework under a Potts model pairwise cost. This assigns a fixed penalty for two neighboring pixels being given different shadow labels. The MRF is optimized using the Tree Reweighted message passing algorithm [14]. Figure 4 shows an example of applying our shadow region segmentation to a real image.

5 Color Photometric-Stereo

It may seem that a photometric stereo scheme with three images is unnecessarily restrictive. The overall cost in practicality of acquiring one more image is small compared to the rest of the process (calibration, darkening the environment, changing the illumination etc). In this section we examine color photometric stereo [4]. This is a setup where it is not possible to obtain more than three images. The key observation is that in an environment where red, green, and blue light is simultaneously emitted from different directions, a Lambertian surface will reflect each of those colors simultaneously without any mixing of the frequencies. The quantities of red, green and blue light reflected are a linear function of the surface normal direction. A color camera can measure these quantities from a single RGB image. Recently [5] it was shown how this idea can be used to obtain a reconstruction of a deforming object. Because color photometric stereo is applied on a single image, one can use it on a video sequence without having to change illumination between frames. In [5] shadowed pixels were detected and discarded. Here we show how to improve that method by incorporating shadow regions into the reconstruction. In color photometric stereo each of the three camera sensors can be seen as one of the three images of classic photometric stereo. The pixel intensity of pixel (x, y) for the i-th sensor is given by

$$c_i(x, y) = \sum_j \left(\mathbf{l}_j^\top \mathbf{n}\right) \int E_j(\lambda) R(x, y, \lambda) S_i(\lambda) d\lambda. \tag{7}$$

Note that now the sensor sensitivity S_i and spectral distribution E_j are different per sensor and per light source respectively. To be able to determine a unique mapping between RGB values and normal orientation we need to assume a monochromatic surface. We therefore require that $R(x, y, \lambda) = \alpha(x, y)\,\rho(\lambda)$. Where $\alpha(x, y)$ is the monochromatic albedo of the surface point and $\rho(\lambda)$ is the characteristic chromaticity of the material. Let

$$v_{ij} = \int E_j(\lambda)\,\rho(\lambda)\,S_i(\lambda)\,d\lambda$$

and

$$\mathbf{v}_j = \begin{pmatrix} v_{1j} & v_{1j} & v_{1j} \end{pmatrix}^\top.$$

Also define the scaled normal to be

$$\mathbf{b} = \alpha(x, y)\,\mathbf{n}.$$

Then the vector of the three sensor responses at a pixel is given by

$$\mathbf{c} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix} \begin{bmatrix} \mathbf{l}_1 & \mathbf{l}_2 & \mathbf{l}_3 \end{bmatrix}^\top \mathbf{b}.$$

Essentially each vector $\mathbf{v}_j$ provides the response measured by the three sensors when a unit of light from source j is received by the camera. If matrix $\begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix}$ is known, then we can compute

$$\hat{\mathbf{c}} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix}^{-1} \mathbf{c}.$$

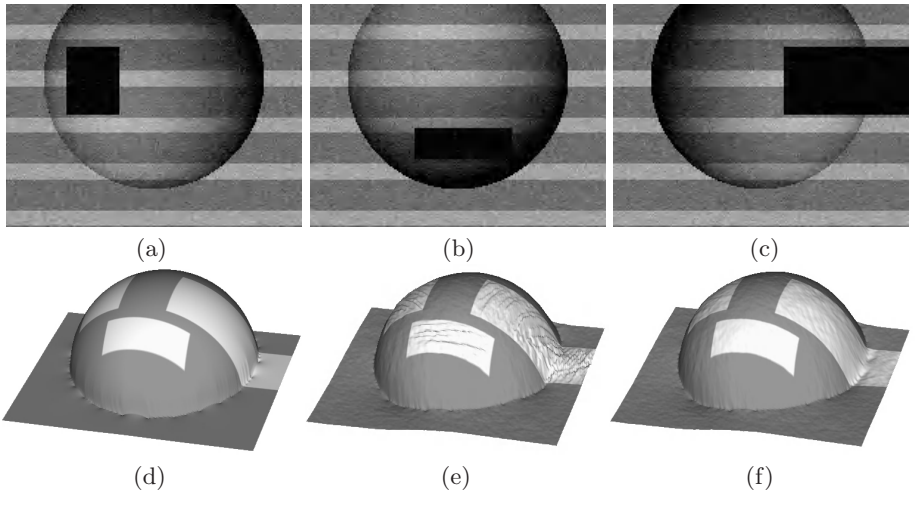


Fig. 2. Sphere sequence. In this experiment we validate our regularization scheme on a synthetic sphere with varying albedo and 10% Gaussian noise. This object is illuminated from three directions with an occluded black region (a-c). The image in (d) shows the reconstruction in the absence of noise and regularization. The image in (e) shows the effect of optimizing the surface using integrability alone. Note the characteristic 'scratch' artifacts. The image in (f) shows the resulting surface after regularization. The artifacts have been suppressed while the data has been preserved unsmoothed.

The values of $\hat{\mathbf{c}}$ can be treated in exactly the same way as the three gray-scale images of section (2). The next section describes a simple process for calibrating color photometric stereo for handling shadows.

5.1 Calibration

In [5] the authors propose a simple scheme for calibrating objects that can be flattened and placed on a planar board. The system detects special patterns on the board, from which it can estimate its orientation relative to the camera. By measuring the RGB response corresponding to each orientation of the material they estimate the entire matrix

$$M = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix} \begin{bmatrix} \mathbf{l}_1 & \mathbf{l}_2 & \mathbf{l}_3 \end{bmatrix}^\top$$

that links the scaled normals to RGB triplets. We propose a two-step process. Firstly, we use a mirror sphere to estimate light directions $\mathbf{l}_1$, $\mathbf{l}_2$ and $\mathbf{l}_3$. This is a standard process which has been applied in a number of photometric stereo methods. Secondly, we capture three sequences of the object moving in front of the camera. In each sequence, we switch on only one of the three lights at a time. In the absence of noise and if the monochromatic assumption was satisfied, the RGB triplets we acquired would be multiples of $\mathbf{v}_j$ when light j was switched on. We therefore do a least squares fit to the three sets of RGB to get the directions of $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$. To get the relative lengths of the three vectors we can use the ratios of the lengths of the RGB vectors. The length of $\mathbf{v}_j$ is set to the maximum length in RGB space, of all the triplets when light j was switched on.

6 Experiments

We present one synthetic experiment and two real experiments on a Frog sequence and a Face sequence.

In figure 2 we study the effect of the proposed framework to automatically detect and correct light occlusions on a half sphere with varying albedo. Figure 2d shows a perfect reconstruction of the sphere in the absence of noise. The algorithm is capable of segmenting the shadowed regions and recovering the shape without any type of regularization on the albedo. However, as soon as we add noise (see Fig. 2e), the recovered shape shows some characteristic artifacts due to an almost unconstrained variation of w in eq.(5) along the shadow line. These artifacts basically show that the recovered shape and albedo are coupled and integrability constraints on their own are not enough to separate them when one intensity constraint is missing. Introducing the regularization term of section 4.1 adds a prior on the intensity of the missing channel. This helps recover the correct shape without loosing any information (see Fig. 2f). In terms of quantitative results, we have compared the reconstructed normal maps with the ground truth sphere in terms of angle difference, the results being as follows:

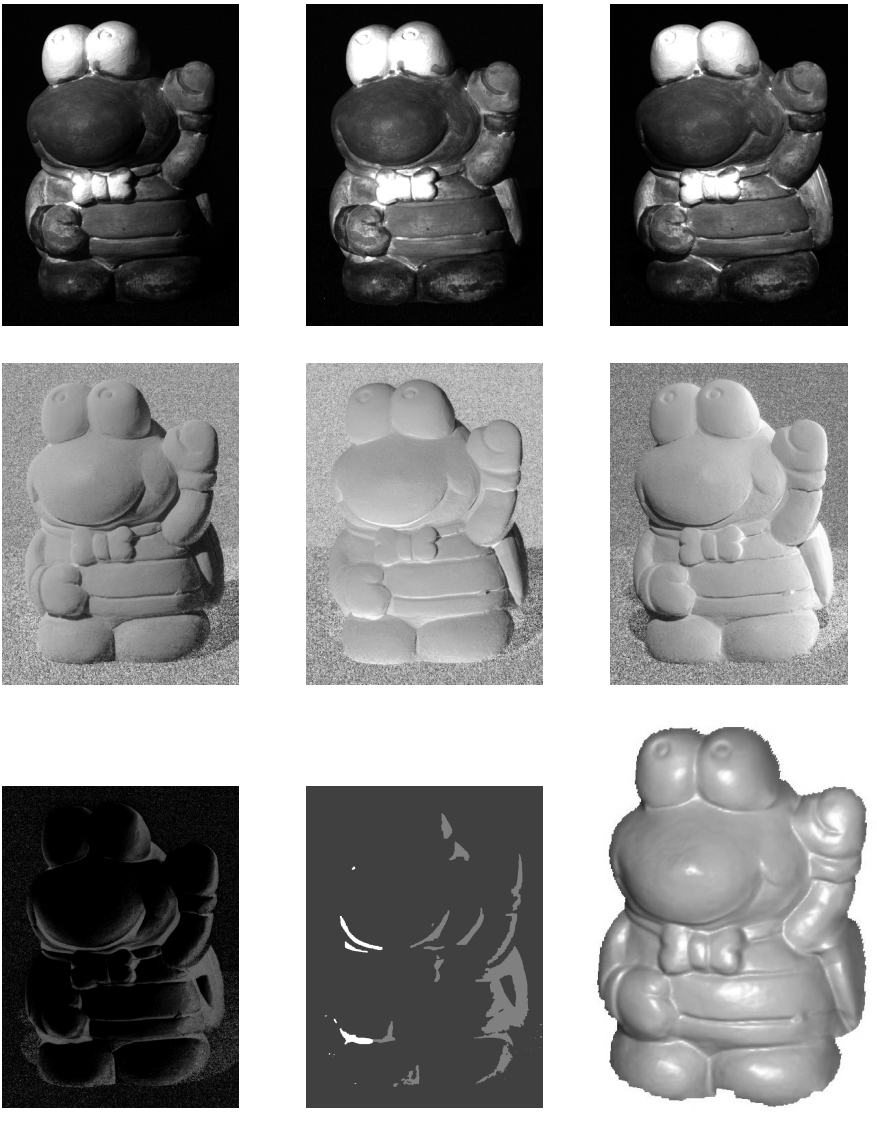


Fig. 3. Frog sequence courtesy of Manmohan Chandraker. In this experiment we use three images of the Frog sequence (top). The middle row shows the normalized images $\frac{c_i}{\|\mathbf{c}\|}$. Note that they do not show any traces of the albedo variation shown in the actual images. In the third row from left to right: The non-shadowing cost $\sqrt{3} - \min_i \frac{c_i}{\|\mathbf{c}\|}$. The shadow segmentation obtained by our proposed scheme. The final surface reconstruction.

algorithm	shadows present	error RMS
ignoring shadows	yes	29.52 degrees
just integrability (Fig. 2e)	yes	15.79 degrees
proposed method (Fig. 2f)	**yes**	**8.67** degrees
data without shadows (ideal case)	no	8.30 degrees

As shown by the table, the proposed method performs almost as well as in the ideal case without any shadows. We obtain an improvement factor of 2 with respect to just using integrability, and a factor of 4 with respect to ignoring the fact that shadows are present.

As a first experimental validation with real data, we use the Frog dataset of [2] which consists of 5 photographs of an object with varying albedo illuminated from 5 different directions. In order to demonstrate the effectiveness of out technique we select a subset of only three images shown in Fig. 3 top. The normalized images $\frac{c_i}{\|\mathbf{c}\|}$ in the middle of Fig. 3 allow the algorithm to easily detect and segment the shadows and to obtain a very accurate shape reconstruction (see Fig. 3 bottom). Note how these images are almost completely invariant to albedo changes in the object surface.

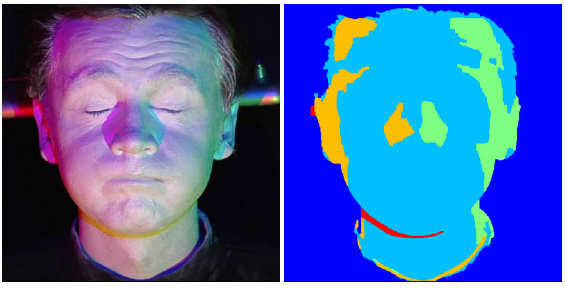


Fig. 4. Shadow segmentation. This experiment shows the result of our shadow region segmentation. On the left is the input image and on the right is the mask with the resulting shadow labels.

Finally, we have performed a second experiment with video data of a white-painted face illuminated by three colored lights in a similar way as in [5] (see supplemental video and Fig. 4 left). The setup is calibrated as described in section 5.1. The automatic shadow segmentation results in Fig. 4 right, demonstrate the accuracy of the shadow detection algorithm in Section 4.2. Figure 5 shows the reconstruction of 3 different frames of the video sequence without taking the shadows into consideration (top) and after detecting and adding the additional constraints to the linear solver (middle). We can appreciate how the nose reconstruction is dramatically improved when correctly processing the shadows, even though only two lights are visible in the shadowed regions (bottom).

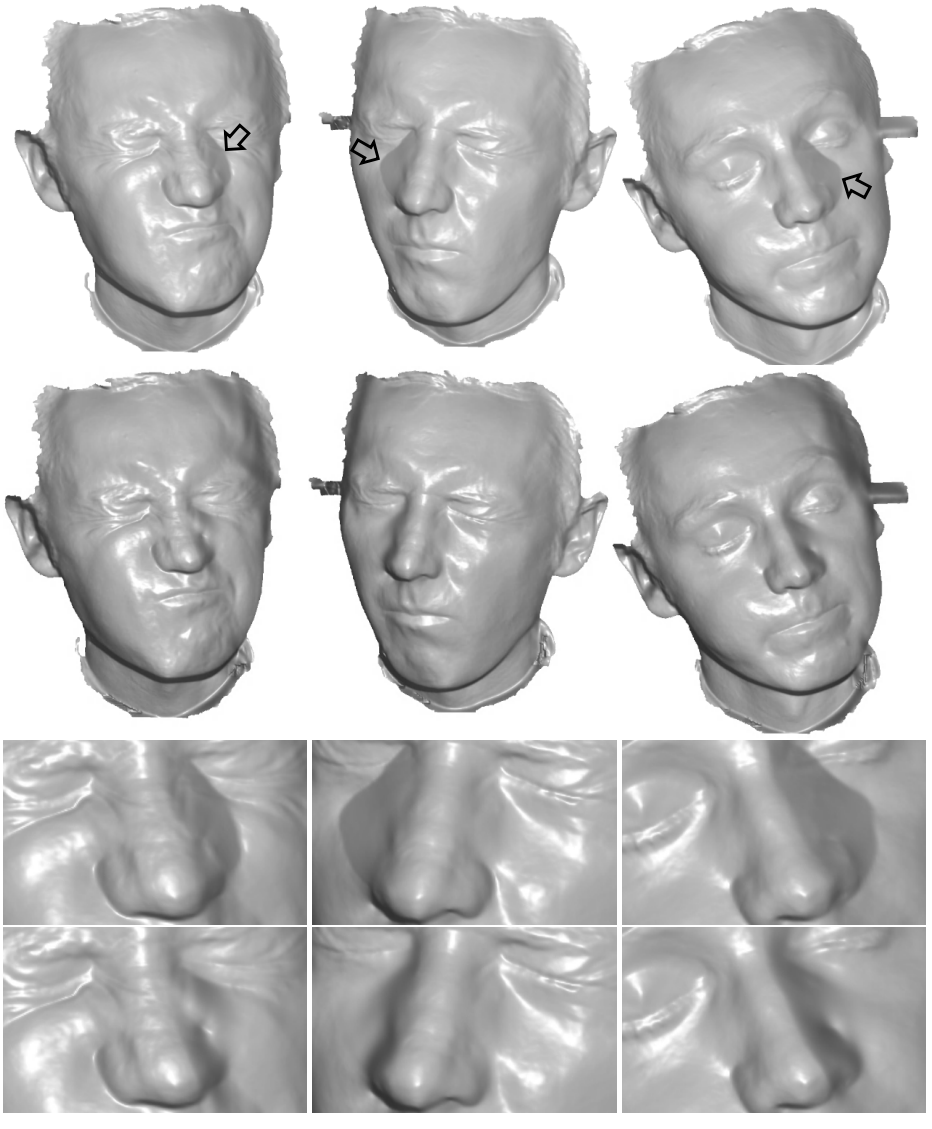


Fig. 5. Face sequence. Three different frames of a video sequence of a face (see supplemental video). The top row shows the reconstruction when shadows are ignored. On the second row are the corresponding reconstructions after detecting and compensating for the shadow regions. Last two rows show a close-up of the face. Note the improvement in the regions around the nose where strong cast shadows appear (see arrows).

7 Discussion

This paper investigated the problem of shadows in the context of three-source photometric stereo. This is a particularly challenging case because the surface

orientation is under-determined inside shadow regions. This is because one of the three necessary constraints is missing due to the shadow. We have shown however that by exploiting integrability, one can still use the remaining two constraints to estimate the surface orientation. In its pure form the problem is ill posed even in the presence of some noise in the data. We provided a remedy to this in the form of a regularization scheme that does not suppress the data of the problem. To detect and segment shadows the paper described a simple MRF optimization scheme, based on the relative darkness of pixel locations. Finally we showed how the ideas in this paper can be applied to the interesting acquisition setup of color photometric stereo.

References

1. Barsky, S., Petrou, M.: The 4-source photometric stereo technique for three-dimensional surfaces in the presence of highlights and shadows. PAMI 25(10), 1239–1252 (2003)
2. Chandraker, M., Agarwal, S., Kriegman, D.: Shadowcuts: Photometric stereo with shadows. In: IEEE Conference on Computer Vision and Pattern Recognition (June 2007)
3. Drew, M.: Reduction of rank-reduced orientation-from-color problem with many unknown lights to two-image known-illuminant photometric stereo. In: IEEE International Symposium on Computer Vision, pp. 419–424 (1995)
4. Petrov, A.: Light, color and shape. Cognitive Processes and their Simulation, 350–358 (1987) (in Russian)
5. Hernández, C., Vogiatzis, G., Brostow, G., Stenger, B., Cipolla, R.: Non-rigid photometric stereo with colored lights. In: Proc. 11^{th} Intl. Conf. on Computer Vision (ICCV) (2007)
6. Woodham, R.: Photometric method for determining surface orientation from multiple images. Optical Engineering 19(1), 139–144 (1980)
7. Lee, S., Brady, M.: Integrating stereo and photometric stereo to monitor the development of glaucoma. Image and Vision Computing 9, 39–44 (1991)
8. Onn, R., Bruckstein, A.: Integrability disambiguates surface recovery in two-image photometric stereo. Int. J. Comput. Vision 5(1), 105 (1990)
9. Coleman Jr., E., Jain, R.: Obtaining 3-dimensional shape of textured and specular surfaces using four-source photometry. Computer Graphics and Image Processing 18(4), 309–328 (1982)
10. Yuille, A., Snow, D.: Shape and albedo from multiple images using integrability. In: CVPR 1997: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR 1997), Washington, DC, USA, p. 158. IEEE Computer Society, Los Alamitos (1997)
11. Tankus, A., Kiryati, N.: Photometric stereo under perspective projection. In: Proc. 10^{th} Intl. Conf. on Computer Vision (ICCV), Washington, DC, USA, pp. 611–616. IEEE Computer Society, Los Alamitos (2005)
12. Andersen, E., Roos, C., Terlaky, T.: On implementing a primal-dual interior-point method for conic quadratic optimization. Math. Prog. 95(2), 249–277 (2003)
13. Agrawal, A., Raskar, R., Chellappa, R.: What is the range of surface reconstructions from a gradient field? In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 578–591. Springer, Heidelberg (2006)
14. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. IEEE Trans. Pattern Anal. Mach. Intell. 28(10), 1568–1583 (2006)

Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search

Herve Jegou, Matthijs Douze, and Cordelia Schmid

INRIA Grenoble, LEAR, LJK
firstname.lastname@inria.fr

Abstract. This paper improves recent methods for large scale image search. State-of-the-art methods build on the bag-of-features image representation. We, first, analyze bag-of-features in the framework of approximate nearest neighbor search. This shows the sub-optimality of such a representation for matching descriptors and leads us to derive a more precise representation based on 1) Hamming embedding (HE) and 2) weak geometric consistency constraints (WGC). HE provides binary signatures that refine the matching based on visual words. WGC filters matching descriptors that are not consistent in terms of angle and scale. HE and WGC are integrated within the inverted file and are efficiently exploited for all images, even in the case of very large datasets. Experiments performed on a dataset of one million of images show a significant improvement due to the binary signature and the weak geometric consistency constraints, as well as their efficiency. Estimation of the full geometric transformation, i.e., a re-ranking step on a short list of images, is complementary to our weak geometric consistency constraints and allows to further improve the accuracy.

1 Introduction

We address the problem of searching for similar images in a large set of images. Similar images are defined as images of the same object or scene viewed under different imaging conditions, cf. Fig. 5 for examples. Many previous approaches have addressed the problem of matching such transformed images [1,2,3,4,5]. They are in most cases based on local invariant descriptors, and either match descriptors between individual images or search for similar descriptors in an efficient indexing structure. Various approximate nearest neighbor search algorithms such as kd-tree [1] or sparse coding with an overcomplete basis set [6] allow for fast search in small datasets. The problem with these approaches is that all individual descriptors need to be compared to and stored.

In order to deal with large image datasets, Sivic and Zisserman [4] introduced the bag-of-features (BOF) image representation in the context of image search. Descriptors are quantized into visual words with the k-means algorithm. An image is then represented by the frequency histogram of visual words obtained by assigning each descriptor of the image to the closest visual word. Fast access to the frequency vectors is obtained by an inverted file system. Note that this approach is an approximation to the direct matching of individual descriptors and

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 304–317, 2008.

somewhat decreases the performance. It compares favorably in terms of memory usage against other approximate nearest neighbor search algorithms, such as the popular Euclidean locality sensitive hashing (LSH) [7,8]. LSH typically requires 100–500 bytes per descriptor to index, which is not tractable, as a one million image dataset typically produces up to 2 billion local descriptors.

Some recent extensions of the BOF approach speed up the assignment of individual descriptors to visual words [5,9] or the search for frequency vectors [10,11]. Others improve the discriminative power of the visual words [12], in which case the entire dataset has to be known in advance. It is also possible to increase the performance by regularizing the neighborhood structure [10] or using multiple assignment of descriptors to visual words [10,13] at the cost of reduced efficiency. Finally, post-processing with spatial verification, a re-occurring technique in computer vision [1], improves the retrieval performance. Such a post-processing was recently evaluated in the context of large scale image search [9].

In this paper we present an approach complementary to those mentioned above. We make the distance between visual word frequency vectors more significant by using a more informative representation. Firstly, we apply a Hamming embedding (HE) to the descriptors by adding binary signatures which refine the visual words. Secondly, we integrate weak geometric consistency (WGC) within the inverted file system which penalizes the descriptors that are not consistent in terms of angle and scale. We also use a-priori knowledge on the transformations for further verification.

This paper is organized as follows. The interpretation of a BOF representation as a an image voting system is given in Section 2. Our contributions, HE and WGC, are described in sections 3 and 4. Complexity issues of our approach in the context of an inverted file system are discussed in Section 5. Finally, Section 6 presents the experimental results.

2 Voting Interpretation of Bag-of-Features

In this section, we show how image search based on BOF can be interpreted as a voting system which matches individual descriptors with an approximate nearest neighbor (NN) search. We then evaluate BOF from this perspective.

2.1 Voting Approach

Given a query image represented by its local descriptors $y_{i'}$ and a set of database images j, $1 \leq i \leq n$, represented by its local descriptors $x_{i,j}$, a voting system can be summarized as follows:

1. Dataset images scores s_j are initialized to 0.
2. For each query image descriptor $y_{i'}$ and for each descriptor $x_{i,j}$ of the dataset, increase the score s_j of the corresponding image by

$$s_j := s_j + f(x_{i,j}, y_{i'}), \tag{1}$$

where f is a matching function that reflects the similarity between descriptors $x_{i,j}$ and $y_{i'}$. For a matching system based on ε-search or $k-$NN, $f(.,.)$ is defined as

$$f_\varepsilon(x,y) = \begin{cases} 1 & \text{if } d(x,y) < \varepsilon \\ 0 & \text{otherwise} \end{cases} \qquad f_{k\text{-NN}}(x,y) = \begin{cases} 1 & \text{if } x \text{ is a } k\text{-NN of } y \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $d(.,.)$ is a distance (or dissimilarity measure) defined on the descriptor space. SIFT descriptors are typically compared using the Euclidean distance.

3. The image score $s_j^* = g_j(s_j)$ used for ranking is obtained from the final s_j. It can formally be written as

$$s_j^* = g_j\left(\sum_{i'=1..m'} \sum_{i=1..m_j} f(x_{i,j}, y_{i'})\right). \tag{3}$$

The simplest choice is $s_j^* = s_j$. In this case the score reflects the number of matches between the query and each database image. Note that this score counts possible multiple matches of a descriptor. Another popular choice is to take into account the number of image descriptors, for example $s_j^* = s_j/m_j$. The score then reflects the rate of descriptors that match.

2.2 Bag-of-Features: Voting and Approximate NN Interpretation

Bag-of-features (BOF) image search uses descriptor quantization. A quantizer q is formally a function

$$\begin{aligned} q : \mathbb{R}^d &\to [1,k] \\ x &\mapsto q(x) \end{aligned} \tag{4}$$

that maps a descriptor $x \in \mathbb{R}^d$ to an integer index. The quantizer q is often obtained by performing k-means clustering on a learning set. The resulting centroids are also referred to as *visual words*. The quantizer $q(x)$ is then the index of the centroid closest to the descriptor x. Intuitively, two descriptors x and y which are close in descriptor space satisfy $q(x) = q(y)$ with a high probability. The matching function f_q defined as

$$f_q(x,y) = \delta_{q(x),q(y)}, \tag{5}$$

allows the efficient comparison of the descriptors based on their quantized index. Injecting this matching function in (3) and normalizing the score by the number of descriptors of both the query image and the dataset image j, we obtain

$$s_j^* = \frac{1}{m_j\ m'} \sum_{i'=1..m'} \sum_{i=1..m_j} \delta_{q(x_{i,j}),q(y_{i'})} = \sum_{l=1..k} \frac{m'_l}{m'} \frac{m_{l,j}}{m_j}, \tag{6}$$

where m'_l and $m_{l,j}$ denote the numbers of descriptors, for the query and the dataset image j, respectively, that are assigned to the visual word l. Note that these scores correspond to the inner product between two BOF vectors. They

are computed very efficiently using an inverted file, which exploits the sparsity of the BOF, i.e., the fact that $\delta_{q(x_{i,j}),q(y_{i'})} = 0$ for most of the (i, j, i') tuples.

At this point, these scores do not take into account the *tf-idf* weighting scheme (see [4] for details), which weights the visual words according to their frequency: rare visual words are assumed to be more discriminative and are assigned higher weights. In this case the matching function f can be defined as

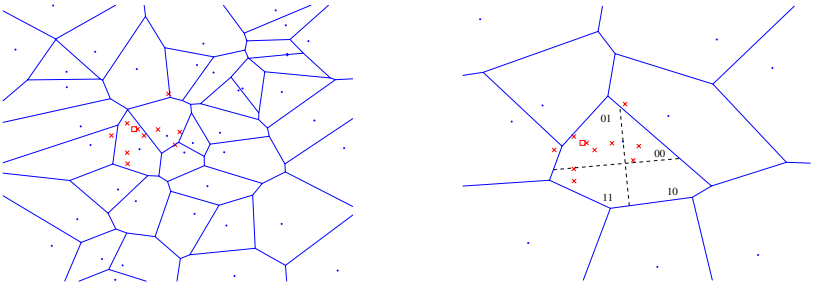
$$f_{\text{tf-idf}}(x, y) = \left(\text{tf-idf}\left(q(y)\right)\right)^2 \delta_{q(x),q(y)}, \tag{7}$$

such that the *tf-idf* weight associated with the visual word considered is applied to both the query and the dataset image in the BOF inner product. Using this new matching function, the image scores s_j become identical to the BOF similarity measure used in [4]. This voting scheme normalizes the number of votes by the number of descriptors (L_1 normalization). In what follows, we will use the L_2 normalization instead. For large vocabularies, the L_2 norm of a BOF is very close to the square root of the L_1 norm. In the context of a voting system, the division of the score by the L_2 norm is very similar to $s_j^* = s_j/\sqrt{m_j}$, which is a compromise between measuring the number and the rate of descriptor matches.

2.3 Weakness of Quantization-Based Approaches

Image search based on BOF combines the advantages of local features and of efficient image comparison using inverted files. However, the quantizer reduces significantly the discriminative power of the local descriptors. Two descriptors are assumed to match if they are assigned the same quantization index, i.e., if they lie in the same Voronoi cell. Choosing the number of centroids k is a compromise between the quantization noise and the descriptor noise.

Fig. 1(b) shows that a low value of k leads to large Voronoi cells: the probability that a noisy version of a descriptor belongs to the correct cell is high.



(a) $k = 40$ (b) $k = 10$

Fig. 1. Illustration of k-means clustering and our binary signature. (a) Fine clustering. (b) Low k and binary signature: the similarity search within a Voronoi cell is based on the Hamming distance. Legend: ·=centroids, □=descriptor, ×=noisy versions of the descriptor.

However, this also reduces the discriminative power of the descriptor: different descriptors lie in the same cell. Conversely, a high value of k provides good precision for the descriptor, but the probability that a noisy version of the descriptor is assigned to the same cell is lower, as illustrated in Fig. 1(a).

We have measured the quality of the approximate nearest neighbor search performed by BOF in terms of the trade-off between (a) the average recall for the ground truth nearest neighbor and (b) the average rate of vectors that match in the dataset. Clearly, a good approximate nearest neighbor search algorithm is expected to make the nearest neighbor vote with high probability, and at the same time arbitrary vectors vote with low probability. In BOF, the trade-off between these two quantities is managed by the number k of clusters. For the evaluation, we have used the approximate nearest neighbor evaluation set available at [14]. It has been generated using the affine covariant features program of [15]. A one million vector set to be searched and a test query set of 10000 vectors are provided. All these vectors have been extracted from the INRIA Holidays image dataset described in Section 6.

One can see in Fig. 2 that the performance of BOF as an approximate nearest neighbor search algorithm is of reasonable accuracy: for $k = 1000$, the NN recall is of 45% and the proportion of the dataset points which are retrieved is of 0.1%. One key advantage of BOF is that its memory usage is much lower than concurrent approximate nearest neighbor search algorithms. For instance, with 20 hash functions the memory usage of LSH [7] is of 160 bytes per descriptors compared to about 4 bytes for BOF. In next section, we will comment on the other curves of Fig. 2, which provide a much better performance than standard BOF.

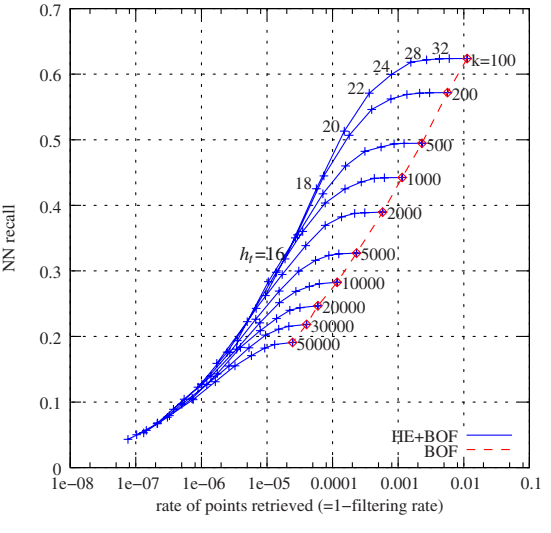


Fig. 2. Approximate nearest neighbor search accuracy of BOF (dashed) and Hamming Embedding (plain) for different numbers of clusters k and Hamming thresholds h_t

3 Hamming Embedding of Local Image Descriptors

In this section, we present an approach which combines the advantages of a coarse quantizer (low number of centroids k) with those of a fine quantizer (high k). It consists in refining the quantized index $q(x_i)$ with a d_b-dimensional binary signature $b(x_i) = (b_1(x_i), \ldots, b_{d_b}(x_i))$ that encodes the localization of the descriptor within the Voronoi cell, see Fig. 1(b). It is designed so that the Hamming distance

$$h(b(x), b(y)) = \sum_{1 \leq i \leq d_b} \delta_{b_i(x), b_i(y)} \tag{8}$$

between two descriptors x and y lying in the same cell reflects the Euclidean distance $d(x, y)$. The mapping from the Euclidean space into the Hamming space, referred to as Hamming Embedding (HE), should ensure that the Hamming distance h between a descriptor and its NNs in the Euclidean space is small.

Note that this significantly different from the Euclidean version of LSH (E2LSH) [7,8], which produces several hash keys per descriptor. In contrast, HE implicitly defines a single partitioning of the feature space and uses the Hamming metric between signatures in the embedded space.

We propose in the following a binary signature generation procedure. We distinguish between 1) the *off-line* learning procedure, which is performed on a learning dataset and generates a set of fixed values, and 2) the binary signature computation itself. The offline procedure is performed as follows:

1. **Random matrix generation:** A $d_b \times d$ orthogonal projection matrix P is generated. We randomly draw a matrix of Gaussian values and apply a QR factorization to it. The first d_b rows of the orthogonal matrix obtained by this decomposition form the matrix P.
2. **Descriptor projection and assignment:** A large set of descriptors x_i from an independent dataset is projected using P. These descriptors $(z_{i1}, ..., z_{id_b})$ are assigned to their closest centroid $q(x_i)$.
3. **Median values of projected descriptors:** For each centroid l and each projected component $h = 1, \ldots, d_b$, we compute the median value $\tau_{l,h}$ of the set $\{z_{ih} | q(x_i) = l\}$ that corresponds to the descriptors assigned to the cell l.

The fixed projection matrix P and $k \times d_b$ median values $\tau_{h,l}$ are used to perform the HE of a given descriptor x by:

1. **Assigning** x to its closest centroid, resulting in $q(x)$.
2. **Projecting** x using P, which produces a vector $z = Px = (z_1, \ldots, z_{d_b})$.
3. **Computing the signature** $b(x) = (b_1(x), \ldots, b_{d_b}(x))$ as

$$b_i(x) = \begin{cases} 1 & \text{if } z_i > \tau_{q(x),i}, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

At this point, a descriptor is represented by $q(x)$ and $b(x)$. We can now define the HE matching function as

$$f_{\mathrm{HE}}(x, y) = \begin{cases} \text{tf-idf}(q(x)) & \text{if } q(x) = q(y) \text{ and } h\left(b(x), b(y)\right) \leq h_t \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

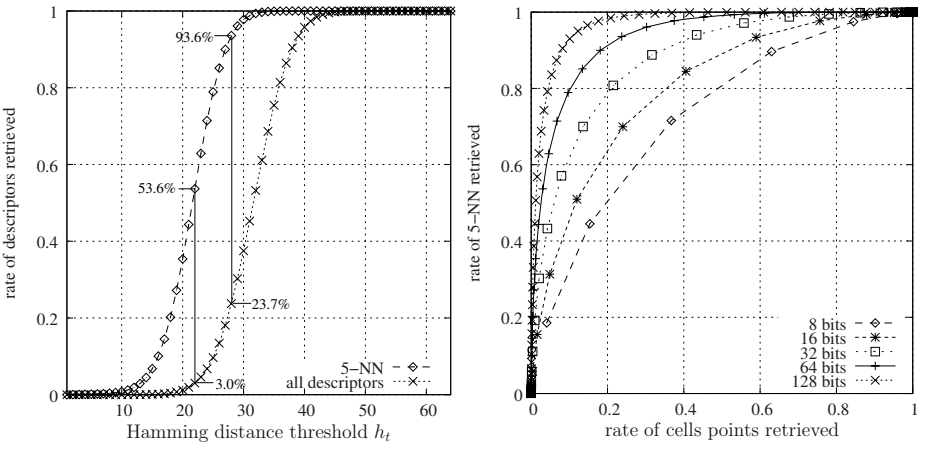


Fig. 3. Filtering effect of HE on the descriptors within a cell and on the 5 NNs. Left: trade-off between the rate of cell descriptors and the rate of NN that are retrieved for $d_b = 64$. Right: impact of the number of bits d_b of the binary signature length.

where h is the Hamming distance defined in Eqn. 9 and h_t is a fixed Hamming threshold such that $0 \leq h_t \leq d_b$. It has to be sufficiently high to ensure that the Euclidean NNs of x match, and sufficiently low to filter many points that lie in a distant region of the Voronoi cell. Fig. 3 depicts this compromise. The plots have been generated by analyzing a set of 1000 descriptors assigned to the same centroid. Given a descriptor x we compare the rate of descriptors that are retrieved by the matching function to the rate of 5-NN that are retrieved.

The left plot shows that the choice of an appropriate threshold h_t (here between 20 and 30) ensures that most of the cell's descriptors are filtered and that the descriptor's NNs are preserved with a high probability. For instance, setting $h_t = 22$ filters about 97% of the descriptors while preserving 53% of the 5-NN. A higher value $h_t = 28$ keeps 94% of the 5-NN and filters 77% of the cell descriptors. Fig. 3(right) represents this trade-off for different binary signature lengths. Clearly, the longer the binary signature d_b, the better the HE filtering quality. In the following, we have fixed $d_b = 64$, which is a good compromise between HE accuracy and memory usage (8 bytes per signature).

The comparison with standard BOF shows that the approximate nearest neighbor search performed by BOF+HE is much better, see Fig. 2. Using HE for the same number of vectors that are retrieved, increases the probability that the NN is among these voting vectors.

4 Large-Scale Geometric Consistency

BOF based image search ranks the database images without exploiting geometric information. Accuracy may be improved by adding a *re-ranking* stage [9] that computes a geometric transformation between the query and a shortlist of

dataset images returned by the BOF search. To obtain an efficient and robust estimation of this transformation, the model is often kept as simple as possible [1,9]. In [1] an affine 2D transformation is estimated in two stages. First, a Hough scheme estimates a transformation with 4 degrees of freedom. Each pair of matching regions generates a set of parameters that "vote" in a 4D histogram. In the second stage, the sets of matches from the largest bins are used to estimate a finer 2D affine transform. In [9] further efficiency is obtained by a simplified parameter estimation and an approximate local descriptor matching scheme.

Despite these optimizations, existing geometric matching algorithms are costly and cannot reasonably be applied to more than a few hundred images. In this section, we propose to exploit weak, i.e., partial, geometrical information without explicitly estimating a transformation mapping the points from an image to another. The method is integrated into the inverted file and can efficiently be applied to all images. Our weak geometric consistency constraints refine the voting score and make the description more discriminant. Note that a *re-ranking* stage [9] can, in addition, be applied on a shortlist to estimate the full geometric transformation. It is complementary to the weak consistency constraints (see Section 6).

4.1 Weak Geometrical Consistency

The key idea of our method is to verify the consistency of the angle and scale parameters for the set of matching descriptors of a given image. We build upon and extend the BOF formalism of (1) by using *several* scores s_j per image. For a given image j, the entity s_j then represents the histogram of the angle and scale differences, obtained from angle and scale parameters of the interest regions of corresponding descriptors. Although these two parameters are not sufficient to map the points from one image to another, they can be used to improve the image ranking produced by the inverted file. This is obtained by modifying the update step of (1) as follows:

$$s_j(\delta_a, \delta_s) := s_j(\delta_a, \delta_s) + f(x_{i,j}, y_{i'}), \tag{11}$$

where δ_a and δ_s are the quantized angle and log-scale differences between the interest regions. The image score becomes

$$s_j^* = g\left(\max_{(\delta_a,\delta_s)} s_j(\delta_a, \delta_s)\right). \tag{12}$$

The motivation behind the scores of (12) is to use angle and scale information to reduce the scores of the images for which the points are not transformed by consistent angles and scales. Conversely, a set of points consistently transformed will accumulate its votes in the same histogram bin, resulting in a high score.

Experimentally, the quantities δ_a and δ_s have the desirable property of being largely independent: computing separate histograms for angle and scale is as

precise as computing the full 2D histogram of (11). In this case two histograms s_j^a and s_j^s are separately updated by

$$\begin{aligned} s_j^a(\delta_a) &:= s_j^a(\delta_a) + f(x_{i,j}, y_{i'}), \\ s_j^s(\delta_s) &:= s_j^s(\delta_s) + f(x_{i,j}, y_{i'}). \end{aligned} \tag{13}$$

The two histograms can be seen as marginal probabilities of the 2D histogram. Therefore, the final score

$$s_j^* = g\left(\min\left(\max_{\delta_a} s_j^a(\delta_a),\ \max_{\delta_s} s_j^s(\delta_s)\right)\right) \tag{14}$$

is a reasonable estimate of the maximum of (12). This approximation will be used in the following. It significantly reduces the memory and CPU requirements. In practice, the histograms are smoothed by a moving average to reduce the angle and log-scale quantization artifacts. Note that the translation could be theoretically included in WGC. However, for a large number of images, the number of parameters should be in fewer than 2 dimensions, otherwise the memory and CPU costs of obtaining the scores would not be tractable.

4.2 Injecting a Priori Knowledge

We have experimentally observed that the repartition of the angle difference δ_a is different for matching and non-matching image pairs: the angle difference for the matching points follows a non-uniform repartition. This is due to the human tendency to shoot either in "portrait" or "landscape" mode. A similar bias is observed for δ_s: image pairs with the same scale ($\delta_s = 0$) are more frequent. We use the orientation and scale priors to weight the entries of our histograms before extracting their maxima. We have designed two different orientation priors: "same orientation" for image datasets known to be shot with the same orientation (i.e. Oxford) and "$\pi/2$ rotation" for more general bases (i.e. Holidays).

5 Complexity

Both HE and WGC are integrated in the inverted file. This structure is usually implemented as an array that associates a list of entries with each visual word. Each entry contains a database image identifier and the number of descriptors of this image assigned to this visual word. The tf-idf weights and the BOF vector norms can be stored separately. The search consists in iterating over the entries corresponding to the visual words in the query image and in updating the scores accordingly.

An alternative implementation consists in storing one entry per descriptor in the inverted list corresponding to a visual word instead of one entry per image. This is almost equivalent for very large vocabularies, because in this case multiple occurrences of a visual word on an image are rare, i.e., it is not necessary to store the number of occurrences. In our experiments, the overall memory usage was

Table 1. Inverted file memory usage and query time per image for a quad-core

descriptor memory usage

image id		21 bits
orientation		6 bits
log-scale		5 bits
binary signature		64 bits
total	WGC	4 bytes
	HE	12 bytes
	WGC+HE	12 bytes

time per query image (Flickr1M dataset)

	$k = 20000$	$k = 200000$
compute descriptors	0.88 s	
quantization + binary signature	0.36 s	0.60 s
search, baseline	2.74 s	0.62 s
search, WGC	10.19 s	2.11 s
search, HE	1.16 s	0.20 s
search, HE+WGC	1.82 s	0.65 s

not noticeably changed by this implementation. This implementation is required by HE and WGC, because additional information is stored per local descriptor.

HE impact on the complexity: For each inverted file entry, we compute the Hamming distance between the signature of the query and that of the database entry. This is done efficiently with a binary `xor` operation. Entries with a distance above h_t are rejected, which avoids the update of image scores for these entries. Note that this occurs for a fair rate of entries, as shown in Fig. 3.

WGC impact on the complexity: WGC modifies the score update by applying (13) instead of (1). Hence, two bins are updated, instead of one for a standard inverted file. The score aggregation as well as histogram smoothing have negligible computing costs. With the tested parameters, see Table 1(left), the memory usage of the histogram scores is 128 floating point values per image, which is small compared with the inverted lists.

Runtime: All experiments were carried out on 2.6 GHz quad-core computers. As the new inverted file contains more information, we carefully designed the size of the entries to fit a maximum 12 bytes per point, as shown in Table 1(left).

Table 1(right) summarizes the average query time for a one million image dataset. We observe that the binary signature of HE has a negligible computational cost. Due to the high rate of zero components of the BOF for a visual vocabulary of $k = 200000$, the search is faster. Surprisingly, HE reduces the inverted file query time. This is because the Hamming distance computation and thresholding is cheaper than updating the scores. WGC reduces the speed, mostly because the histograms do not fit in cache memory and their memory access pattern is almost random. Most interestingly the search time of HE + WGC is comparable to the inverted file baseline. Note that for $k = 200000$ visual words, the assignment uses a fast approximate nearest neighbor search, i.e., the computation is not ten times slower than for $k = 20000$, which here uses exhaustive search.

6 Experiments

We perform our experiments on two annotated datasets: our own *Holidays* dataset, see Fig. 5, and the Oxford5k dataset. To evaluate large scale image search we also introduce a distractor dataset downloaded from Flickr. For evaluation we

use mean average precision (mAP) [9], i.e., for each query image we obtain a precision/recall curve, compute its average precision and then take the mean value over the set of queries. Descriptors are obtained by the Hessian-Affine detector and the SIFT descriptor, using the software of [15] with the default parameters. Clustering is performed with k-means on the independent Flickr60k dataset. The number of clusters is specified for each experiment.

6.1 Datasets

In the following we present the different datasets used in our experiments.

Holidays *(1491 images, 4.456M descriptors, 500 queries).* We have collected a new dataset which mainly contains personal holiday photos. The remaining ones were taken on purpose to test the robustness to various transformations: rotations, viewpoint and illumination changes, blurring, etc. The dataset includes a very large variety of scene types (natural, man-made, water and fire effects, etc) and images are of high resolution. The dataset contains 500 image groups, each of which represents a distinct scene. The first image of each group is the query image and the correct retrieval results are the other images of the group. The dataset is available at [14].

Oxford5k *(5062 images, 4.977M descriptors, 55 queries).* We also used the Oxford dataset [9]. The images represent Oxford buildings. All the dataset images are in "upright" orientation because they are displayed on the web.

Flickr60k *(67714 images, 140M descriptors)* and **Flickr1M** *(1M images, 2072M descriptors).* We retrieved arbitrary images from Flickr and built two distinct sets: Flickr60k is used to learn the quantization centroids and the HE parameters (median values). For these tasks we have used respectively 5M and 140M descriptors. Flickr1M are distractor images for large scale image search. Compared to *Holidays*, the Flickr datasets are slightly biased, because they include low-resolution images and more photos of humans.

Table 2. Results for *Holidays* and *Oxford* datasets. mAP scores for the baseline, HE, WGC and HE+WGC. Angle prior: same orientation for *Oxford*, $0, \pi/2, \pi$ and $3\pi/2$ rotations for *Holidays*. Vocabularies are generated on the independent Flickr60K dataset.

	Parameters		Holidays		Oxford	
	HE: h_t	WGC	$k = 20000$	$k = 200000$	$k = 20000$	$k = 200000$
baseline			0.4463	0.5488	0.3854	0.3950
HE	20		0.7268	0.7093	0.4798	0.4503
HE	22		0.7181	0.7074	0.4892	0.4571
HE	24		0.6947	0.7115	0.4906	0.4585
HE	26		0.6649	0.6879	0.4794	0.4624
WGC		no prior	0.5996	0.6116	0.3749	0.3833
WGC		with prior	0.6446	0.6859	0.4375	0.4602
HE+WGC	20	with prior	0.7391	0.7328	0.5442	0.5096
HE+WGC	22	with prior	0.7463	0.7382	0.5472	0.5217
HE+WGC	24	with prior	0.7507	0.7439	0.5397	0.5252
HE+WGC	26	with prior	0.7383	0.7404	0.5253	0.5275

Impact of the clustering learning set. Learning the visual vocabulary on a distinct dataset shows more accurately the behavior of the search in very large image datasets, for which 1) query descriptors represent a negligible part of the total number of descriptors, and 2) the number of visual words represents a negligible fraction of the total number of descriptors. This is confirmed by comparing our results on Oxford to the ones of [9], where clustering is performed on the evaluation set. In our case, i.e., for a distinct visual vocabulary, the improvement between a small and large k is significantly reduced when compared to [9], see first row of Table 2.

6.2 Evaluation of HE and WGC

INRIA Holidays and Oxford building datasets: Table 2 compares the proposed methods with the standard BOF baseline. We can observe that both HE and WGC result in significant improvements. Most importantly, the combination of the two further increases the performance.

Large scale experiments: Fig. 4 shows an evaluation of the different approaches for large datasets, i.e., we combined the Holidays dataset with a varying number of images from the 1M Flickr dataset. We clearly see that the gain of the variant WGC + HE is very significant. In the case of WGC + HE the corresponding curves degrade less rapidly when the number of images in the database increases. Results for various queries are presented in Fig. 5. We can observe that HE and WGC improve the quality of the ranking significantly. Table 3 measures this improvement. It gives the rate of true positives that are in a shortlist of 100 images. For a dataset of one million images, the baseline only

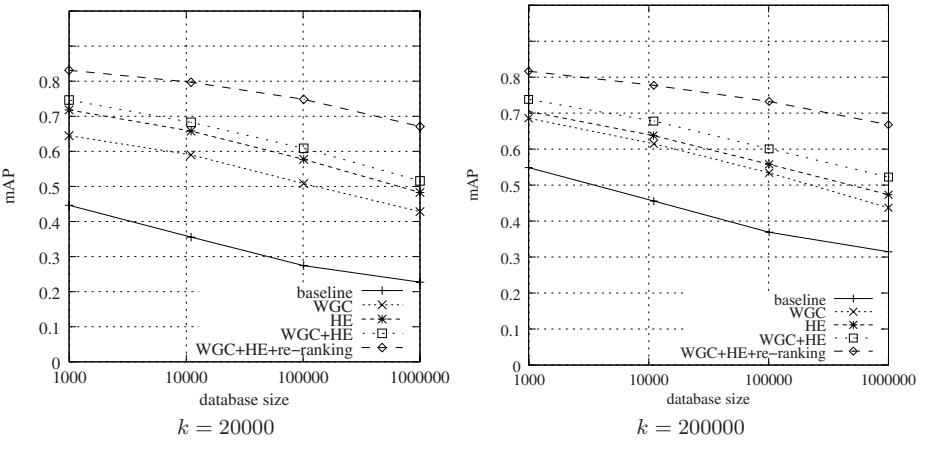


Fig. 4. Performance of the image search as a function of the dataset size for BOF, WGC, HE ($h_t = 22$), WGC+HE, and WGC+HE+re-ranking with a full geometrical verification (shortlist of 100 images). The dataset is *Holidays* with a varying number of distractors from *Flickr1M*.

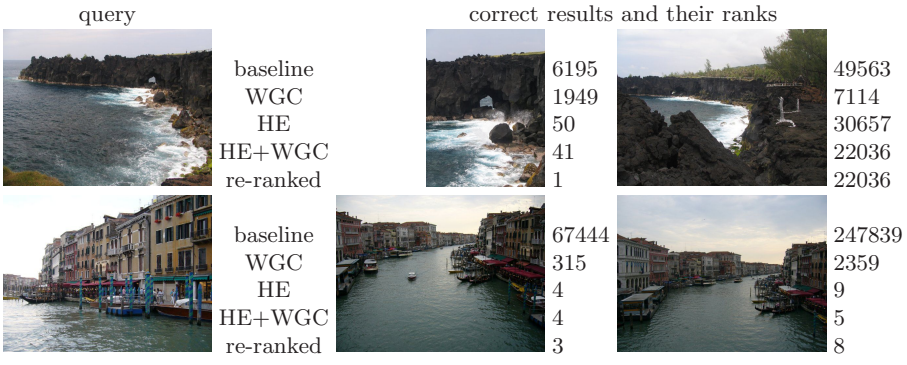


Fig. 5. Queries from the *Holidays* dataset and some corresponding results for *Holidays*+1M distractors from Flickr1M

Table 3. *Holidays dataset + Flickr1M:* Rate of true positives as a function of the dataset size for a shortlist of 100 images, $k = 200000$

dataset size	991	10991	100991	1000991
BOF	0.673	0.557	0.431	0.306
WGC+HE	0.855	0.789	0.708	0.618

returns 31% of the true positive, against 62% for HE+WGC. This reflects the quality of the shortlist that will be considered in a re-ranking stage.

Re-ranking: The re-ranking is based on the estimation of an affine transformation with our implementation of [1]. Fig. 4 also shows the results obtained with a shortlist of 100 images. We can observe further improvement, which confirms the complementary of this step with WGC.

7 Conclusion

This paper has introduced two ways of improving a standard bag-of-features representation. The first one is based on a Hamming embedding which provides binary signatures that refine visual words. It results in a similarity measure for descriptors assigned to the same visual word. The second is a method that enforces weak geometric consistency constraints and uses a priori knowledge on the geometrical transformation. These constraints are integrated within the inverted file and are used for all the dataset images. Both these methods improve the performance significantly, especially for large datasets. Interestingly, our modifications do not result in an increase of the runtime.

Acknowledgments. We would like to acknowledge the ANR projects GAIA and RAFFUT as well as GRAVIT for their financial support.

References

1. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
2. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. IJCV 60(1), 63–86 (2004)
3. Matas, J., Chum, O., Martin, U., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: BMVC, pp. 384–393 (2002)
4. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: ICCV, pp. 1470–1477 (2003)
5. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: CVPR, pp. 2161–2168 (2006)
6. Omercevic, D., Drbohlav, O., Leonardis, A.: High-dimensional feature matching: employing the concept of meaningful nearest neighbors. In: ICCV (2007)
7. Datar, M., Immorlica, N., Indyk, P., Mirrokni, V.: Locality-sensitive hashing scheme based on p-stable distributions, pp. 253–262 (2004)
8. Shakhnarovich, G., Darrell, T., Indyk, P.: Nearest-Neighbor Methods in Learning and Vision: Theory and Practice. MIT Press, Cambridge (2006)
9. Philbin, J., Chum, O., Isard, M.A., Zisserman, J.S.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR (2007)
10. Jegou, H., Harzallah, H., Schmid, C.: A contextual dissimilarity measure for accurate and efficient image search. In: CVPR (2007)
11. Fraundorfer, F., Stewenius, H., Nister, D.: A binning scheme for fast hard drive based image search. In: CVPR (2007)
12. Schindler, G., Brown, M., Szeliski, R.: City-scale location recognition. In: CVPR (2007)
13. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: CVPR (2008)
14. Jegou, H., Douze, M.: INRIA Holidays dataset (2008), `http://lear.inrialpes.fr/people/jegou/data.php`
15. Mikolajczyk, K.: Binaries for affine covariant region descriptors (2007), `http://www.robots.ox.ac.uk/~vgg/research/affine/`

Estimating Geo-temporal Location of Stationary Cameras Using Shadow Trajectories

Imran N. Junejo[1,⋆] and Hassan Foroosh[2]

[1] INRIA Rennes, France
[2] University of Central Florida, Orlando, U.S.A.

Abstract. Using only shadow trajectories of stationary objects in a scene, we demonstrate that using a set of six or more photographs are sufficient to accurately calibrate the camera. Moreover, we present a novel application where, using only three points from the shadow trajectory of the objects, one can accurately determine the geo-location of the camera, up to a longitude ambiguity, and also the date of image acquisition without using any GPS or other special instruments. We refer to this as "geo-temporal localization". We consider possible cases where ambiguities can be removed if additional information is available. Our method does not require any knowledge of the date or the time when the pictures are taken, and geo-temporal information is recovered directly from the images. We demonstrate the accuracy of our technique for both steps of calibration and geo-temporal localization using synthetic and real data.

1 Introduction

Cameras are everywhere. Groups, individuals or governments mount cameras for various purposes like performing video surveillance, observing natural scenery, or for observing weather patterns. As a result, a global network of thousands of outdoor or indoor cameras currently exists on the internet, which provides a flexible and economical method for information sharing. For such a network, the ability to determine geo-temporal information directly from visual cues has a tremendous potential, in terms of applications, for the field of forensics, intelligence, security, and navigation, to name a few.

The cue that we use for *geo-temporal* localization of the camera, (defined henceforth as *the physical location of the camera (GPS coordinates) and the date of image acquisition*) is the shadow trajectories of two stationary objects during the course of a day. The use of shadow trajectory of a gnomon to measure time in a sundial is reported as early as 1500 BC by Egyptians, which surprisingly requires sophisticated astronomical knowledge [1,2,3]. Shadows have been used in multiple-view geometry in the past to provide information about the shape and the 3-D structure of the scene [4,5], or to recover camera intrinsic and extrinsic parameters [6,7]. Determining the GPS coordinates and the date of the year from shadows in images is a new concept that we introduce in this paper.

⋆ This is part of the author's work at the University of Central Florida, Orlando, U.S.A.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 318–331, 2008.

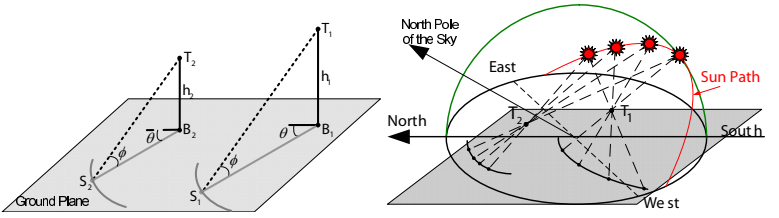


Fig. 1. Two objects T_1 and T_2 casting shadow on the ground plane. The locus of shadow positions over the course of a day is a function of the sun altitude ϕ, the sun azimuth θ and the height h_i of the object.

Our approach is a two step process: auto-calibration and geo-temporal localization. In terms of calibration, the most related work to ours are those of Cao and Foroosh [8] and Lu et al. [9]. The authors in [8] use multiple views of objects and their cast shadows for camera calibration, requiring the objects that cast shadows to be visible in each image and typically from parallel objects perpendicular to the ground plane. Similarly, [9] use line segments formed by corresponding shadow points to estimate the horizon line for camera calibration. Here our contribution is twofold: 1- develop a more flexible solution by relaxing the requirement that shadow-casting objects have to be visible or of particular geometry; 2- provide a more robust solution to estimating the vanishing line of the ground plane by formulating it as a largely overdetermined problem in a manner somewhat similar to [10]. Therefore, our auto-calibration method does not exploit camera motion as in [11,12,13] but rather uses shadows to deduce scene structures that constrain the geometric relations in the image plane [14,15,16].

For geo-temporal localization, recently Jacobs et al.[17] used a database of images collected over a course of a year to learn weather patterns. Using these natural variations, the camera is then geo-located by the correlation of camera images to geo-registered satellite images and also by correlating acquired images with known landmarks/locations. In contrast, the present work is based solely on astronomical geometry and is more flexible, requiring only three shadow points for GPS coordinates estimation. To demonstrate the power of the proposed method we downloaded some images from online traffic surveillance webcams, and determined accurately the geo-locations and the date of acquisition.

Overall two main contributions are made in this paper: *First*, we present a camera calibration method where the horizon line is extracted solely from shadow trajectories without requiring the objects to be visible: we discuss two possible cases (see below). *Second*, we present an innovative application to estimate the GPS coordinates (up to longitude ambiguity) of the location where the images were taken, along with the day of year when the images were taken (up to year ambiguity). In this step, only three points on the shadow trajectories are required, leading to a robust geo-temporal localization. Accordingly, this paper is divided into corresponding sections addressing each issue.

2 Preliminaries and the Setup

Let $\mathbf{T}$ be a 3D stationary point and $\mathbf{B}$ its footprint (i.e. its orthogonal projection) on the ground plane. As depicted in Fig. 1, the locus of shadow positions $\mathbf{S}$ cast by $\mathbf{T}$ on

the ground plane, i.e. the shadow trajectory, is a smooth curve that depends only on the altitude (ϕ) and the azimuth angles (θ) of the sun in the sky and the vertical distance h of the object from its footprint.

Without loss of generality, we take the ground plane as the world plane $z = 0$, and define the x-axis of the world coordinate frame toward the true north point, where the azimuth angle is zero. Therefore, algebraically, the 3D coordinates of the shadow position can be unambiguously specified by their 2D coordinates in the ground plane as

$$\bar{\mathbf{S}}_i = \bar{\mathbf{B}}_i + h_i \cot\phi \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}, \tag{1}$$

where $\bar{\mathbf{S}}_i = [S_{ix} \;\; S_{iy}]^T$ and $\bar{\mathbf{B}}_i = [B_{ix} \;\; B_{iy}]^T$ are the inhomogeneous coordinates of the shadow position $\mathbf{S}_i$, and the object's footprint $\mathbf{B}_i$ on the ground plane. (1) is based on the assumption that the sun is distant and therefore its rays, e.g. $\mathbf{T}_i\mathbf{S}_i$, are parallel to each other. It follows that the shadows $\mathbf{S}_1$ and $\mathbf{S}_2$ of any two stationary points $\mathbf{T}_1$ and $\mathbf{T}_2$ are related by a rotation-free 2D similarity transformation as $\mathbf{S}_2 \sim \mathbf{H}_s^{12}\mathbf{S}_1$, where

$$\mathbf{H}_s^{12} \sim \begin{bmatrix} h_2/h_1 & 0 & B_{2x} - B_{1x}h_2/h_1 \\ 0 & h_2/h_1 & B_{2y} - B_{1y}h_2/h_1 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Note that the above relationship is for world shadow positions and valid for any day time.

Shadow Detection and Tracking: In order to estimate the shadow trajectories, we adopt a semi-automatic approach. For a set of images $\mathbf{S}_I = \{\mathbf{I}_1, \mathbf{I}_2, ...\mathbf{I}_m\}$, we construct a background image $\mathbf{I}$ where each pixel (x, y) contains the brightest pixel value from our set of images $\mathbf{S}_I$. After background subtraction, the most prominent shadow points are detect manually. Mean Shift [18] tracking algorithm is then applied to track the shadow points in the subsequent frames.

3 Recovering the Vanishing Line

Once the vanishing line ($\mathbf{l}_\infty$) is recovered, it can be used together with the vertical vanishing point ($\mathbf{v}_z$), found by fitting lines to vertical directions, to recover the image of the absolute conic (IAC), which is decomposed into the camera calibration matrix $\mathbf{K}$ by using the Cholesky decomposition [19]. For recovering $\mathbf{l}_\infty$, there are two cases that need to be considered:

3.1 When Shadow Casting Object Is Visible

A situation may occur when the footprint, and optionally the shadow casting object itself are visible in the image. An example of this case is the light pole visible in image sequence shown in Figure 6. From here on, the quantities $\mathbf{T}_i, \mathbf{B}_i, \mathbf{S}_i$ and $\mathbf{S}_i'$ refer to the points projected on to the image plane.

Figure 2 illustrates the general setup for this case. The vertical vanishing point is obtained by $\mathbf{v}_z = (\mathbf{T_1} \times \mathbf{B_1}) \times (\mathbf{T_2} \times \mathbf{B_2})$. The estimation of $\mathbf{l}_\infty$ is as follows: at

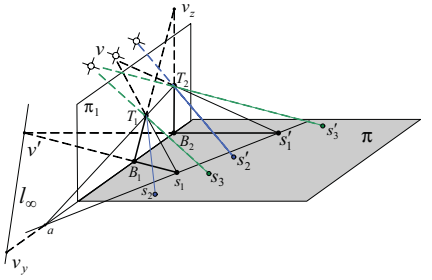


Fig. 2. The setup used for camera calibration and for estimating geo-temporal information

time instance $t = 1$, the sun located at vanishing point $\mathbf{v}_1$ casts shadow of $\mathbf{T}_1$ and $\mathbf{T}_2$ at points $\mathbf{S}_1$ and $\mathbf{S'}_1$, respectively. The sun is a distant object and therefore its rays, $\mathbf{T}_1\mathbf{S}_1$ and $\mathbf{T}_2\mathbf{S'}_1$, are parallel to each other. It then follows that the shadow rays, i.e. $\mathbf{S}_1\mathbf{B}_1$ and $\mathbf{S'}_1\mathbf{B}_2$, are also parallel to each other. These rays intersect at a vanishing point $\mathbf{v}'_1$ on the ground plane. Similarly, for time instance $t = 2$ and $t = 3$, we obtain the vanishing points $\mathbf{v}'_2$ and $\mathbf{v}'_3$, respectively. These vanishing points all lie on the vanishing line of the ground plane on which the shadows are cast, i.e. ${\mathbf{v}'_i}^T\mathbf{l}_\infty = 0$, where $i = 1, 2, \ldots n$ and n is number of instances for which shadow is being observed. Thus a minimum of two observations are required to obtain the $\mathbf{l}_\infty$.

3.2 When Shadow Casting Object Is NOT Visible

This is a more *general* case. The footprint and/or the the shadow casting object point might not always be visible in a video sequence. Figure 7 shows a picture of downtown Washington D.C, where one of the shadow casting objects is the traffic light hanging by a horizontal pole (or a cable). The footprint of this traffic light on the ground plane cannot be determined. In this setup, $\mathbf{l}_\infty$ can not be recovered as described in the previous case. Although, the vertical vanishing point can be obtained by other vertical structures in the scene, not necessarily the shadow-casting structures.

Note: In this case, we use only shadow trajectories to recover the horizon line $\mathbf{l}_\infty$. However, as described in Section 4, we do require to see the shadow casting object (although, not its footprint), in order to perform geo-temporal localization.

Assume now that we have two world points $\mathbf{T}_1$ and $\mathbf{T}_2$ that cast shadows on the ground plane. Given any five imaged shadow positions of the same 3D points ($\mathbf{T}_1$ or $\mathbf{T}_2$), cast at distinct times during one day, one can fit a conic through them, which would meet the line at infinity of the ground plane at two points. These points may be real or imaginary depending on whether the resulting conic is an ellipse, a parabola, or a hyperbola [19]. The two distinct and unique image conics $\mathbf{C}_1$ and $\mathbf{C}_2$ are related by $\mathbf{C}_2 \sim (\mathbf{H}\mathbf{H}_s^{12}\mathbf{H}^{-1})^{-T}\mathbf{C}_1(\mathbf{H}\mathbf{H}_s^{12}\mathbf{H}^{-1})^{-1}$, where $\mathbf{H}$ is the world to image planar homography with respect to the ground plane.

Since the two world conics are similar, owing to the distance of the sun from the observed objects, these two conics generally intersect at *four* points, two of which must lie on the image of the horizon line of the ground plane. The basic idea of conic intersection is illustrated in Fig. 3, for details see Appendix A. It then follows that for any conic $\mathbf{C}_\mu$ through these points of intersection we have (we omit proof here):

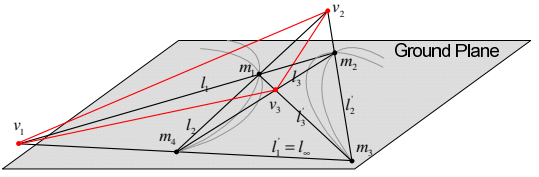


Fig. 3. The two gray conics are fitted by two sets of five distinct shadow positions on the ground plane cast by two world points. Generally, the two conics intersect at four points $\mathbf{m}_i, i = 1, ..., 4$. The diagonal triangle $\Delta\mathbf{v}_1\mathbf{v}_2\mathbf{v}_3$ is self-polar.

Theorem 1. *(Self-Polar Triangle)*
Let $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3$ *and* $\mathbf{m}_4$ *be four points on the conic locus* $\mathbf{C}_\mu$*, the diagonal triangle of the quadrangle* $\mathbf{m}_1\mathbf{m}_2\mathbf{m}_3\mathbf{m}_4$ *is self-polar for* $\mathbf{C}_\mu$*. Since two of the points lie on* $\mathbf{l}_\infty$*, one of the vertices of* $\Delta\mathbf{v}_1\mathbf{v}_2\mathbf{v}_3$ *also lies on* $\mathbf{l}_\infty$*.*

In simple terms, for the two conics $\mathbf{C}_1$ and $\mathbf{C}_2$, two of the four points of intersection $(\mathbf{m}_1, ..., \mathbf{m}_4)$ lie on the horizon line $\mathbf{l}_\infty$. These points of intersection also define a self-polar triangle $\Delta\mathbf{v}_1\mathbf{v}_2\mathbf{v}_3$, one vertex of which also lies on $\mathbf{l}_\infty$ (cf. Fig.3).

Robust estimation of $\mathbf{l}_\infty$: Five points are required to uniquely define a conic. From two such conics ($\mathbf{C}_1$ and $\mathbf{C}_2$), we get three vanishing points (2 from intersection points and one vertex of the self-polar triangle) that lie on $\mathbf{l}_\infty$. Therefore given six or more corresponding image points on the shadow paths of the two objects, we can get six or more self-polar triangles, from which the horizon line of the ground plane can be recovered. Since, two of intersection points (2 points of the quadrangle) are also on the horizon line of the ground plane, they can be used together with one vertex of each self-polar triangle to recover the horizon line. As an example, Figure 4 illustrates the horizon line fitted to many points obtained during our experimentation. If n such vanishing points are available, we define a matrix M such that:

$$\begin{bmatrix} \mathbf{v}_1'^T \\ \mathbf{v}_2'^T \\ \vdots \\ \mathbf{v}_n'^T \end{bmatrix} \mathbf{l}_\infty = \mathbf{M}\mathbf{l}_\infty = 0$$

where the least square solution to this system of equations is the line at infinity, passing though all the vanishing points $\mathbf{v}_i'^T$.

Note that for $n \geq 6$ corresponding points on shadow paths of two objects, we obtain a total of $\frac{3n!}{(n-5)!5!}$ vanishing points. For instance, with only 10 corresponding shadow points, we would get 756 points on the horizon line. This would allow us to very accurately estimate the horizon line in the presence of noise.

3.3 Camera Calibration

In the previous section, we described a novel technique to recover the vanishing line (or the line at infinity) from using only shadow trajectories. $\mathbf{l}_\infty$, together with the vertical

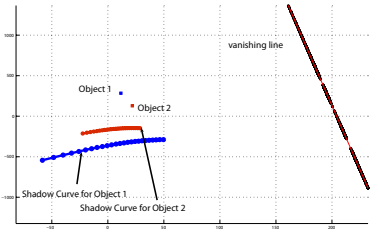


Fig. 4. The horizon line detected from a sequence of self-polar triangles and the intersection of the conics fit on shadow trajectories of two objects

vanishing point $\mathbf{v}_z$, fitted from vertical objects, provide two constraints on the image of the absolute conic (IAC) from pole-polar relationship [19]. Assuming a camera with zero skew, and unit aspect ratio [16,13], the IAC is of the form

$$\boldsymbol{\omega} \sim [\omega_1\ \omega_2\ \omega_3] \sim \begin{bmatrix} 1 & 0 & \omega_{13} \\ 0 & 1 & \omega_{23} \\ \omega_{13} & \omega_{23} & \omega_{33} \end{bmatrix} \tag{3}$$

Let $\mathbf{l}_\infty = \begin{bmatrix} l_x\ l_y\ 1 \end{bmatrix}^T$ and $\mathbf{v}_z = \begin{bmatrix} \mathbf{v}_z^x\ \mathbf{v}_z^y\ 1 \end{bmatrix}^T$, we can solve the linear constraints obtained from the pole-polar relationship to solve for w_{13} and w_{23} in terms of w_{33}:

$$w_{13} = \frac{l_x\, \mathbf{v}_z^{y2} - l_x\, w_{33} - l_y\, \mathbf{v}_z^y\, \mathbf{v}_z^x + \mathbf{v}_z^x}{l_y\, \mathbf{v}_z^y + l_x\, \mathbf{v}_z^x - 1} \tag{4}$$

$$w_{23} = -\frac{l_x\, \mathbf{v}_z^x\, \mathbf{v}_z^y - \mathbf{v}_z^{x\,2} l_y - \mathbf{v}_z^y + l_y\, w_{33}}{l_y\, \mathbf{v}_z^y + l_x\, \mathbf{v}_z^x - 1} \tag{5}$$

The remaining parameter w_{33} is estimated by minimizing the "closeness to the center" constraint [20]:

$$\hat{\omega}_{33} = \arg\min \| \begin{bmatrix} \omega_{13}\ \omega_{23} \end{bmatrix}^T - \mathbf{c} \| \tag{6}$$

where $\mathbf{c}$ is the center of the image, and $\hat{\omega}_{33}$ is the optimal solution for ω_{33}, from which the other two parameters are computed to completely recover the IAC in (3). We use Levenberg-Marquardt to minimize this equation. As most modern cameras have principal points close to the center of the image [16,8], the image center is used as the initial starting point of the minimization process.

4 The Geo-temporal Localization Step

Once we have calibrated the camera, then in order to perform geo-temporal localization, we need to estimate the azimuth and the altitude angle of the sun. At any time of the year, the exact location of the sun can be determined by these two angles. For this it is necessary that the world point casting the shadow on the ground plane be visible in the image.

The earth orbits the sun approximately every 365 days while it also rotates on its axis that extends from the north pole to the south pole every 24 hours. The orbit around the sun is elliptical in shape, which causes it to speed up and slow down as it moves around the sun. The polar axis also tilts up to a maximum angle of about 23.47° with the orbital plane over the course of a year. This tilt causes a change in the angle that the sun makes with the equatorial plane, the so called *declination angle*. Similarly, the globe may be partitioned in several ways. A circle passing through both poles is called a *Meridian*. Another circle that is equidistance from the north and the south pole is called the *Equator*. *Longitude* is the angular distance measured from the prime meridian through Greenwich, England. Similarly, *Latitude* is the angular distance measured from the equator, North $(+ve)$ or South $(-ve)$. Latitude values are important as they define the relationship of a location with the sun. Also, the path of the sun, as seen from the earth, is unique for each latitude, which is the main cue which allows us to geo-locate the camera from only shadow trajectories. Next, we describe the methods for determining these quantities.

Latitude: An overview of the proposed method is shown in Fig. 2. Let $\mathbf{s}_i$, $i = 1, 2, 3$ be the images of the shadow points of a stationary object recorded at different times during the course of a single day. Let $\mathbf{v}_i$ and $\mathbf{v'}_i$, $i = 1, 2, 3$ be the sun and the shadow vanishing points, respectively. For a calibrated camera, the following relations hold for the altitude angle ϕ_i and the azimuth angle θ_i of the sun orientations in the sky, all of which are measured directly in the image domain:

$$\cos\phi_i = \frac{\mathbf{v'}_i^T \boldsymbol{\omega} \mathbf{v}_i}{\sqrt{\mathbf{v'}_i^T \boldsymbol{\omega} \mathbf{v'}_i}\sqrt{\mathbf{v}_i^T \boldsymbol{\omega} \mathbf{v}_i}} \tag{7}$$

$$\sin\phi_i = \frac{\mathbf{v}_z^T \boldsymbol{\omega} \mathbf{v}_i}{\sqrt{\mathbf{v}_z^T \boldsymbol{\omega} \mathbf{v}_z}\sqrt{\mathbf{v}_i^T \boldsymbol{\omega} \mathbf{v}_i}} \tag{8}$$

$$\cos\theta_i = \frac{\mathbf{v}_y^T \boldsymbol{\omega} \mathbf{v'}_i}{\sqrt{\mathbf{v}_y^T \boldsymbol{\omega} \mathbf{v}_y}\sqrt{\mathbf{v'}_i^T \boldsymbol{\omega} \mathbf{v'}_i}} \tag{9}$$

$$\sin\theta_i = \frac{\mathbf{v}_x^T \boldsymbol{\omega} \mathbf{v'}_i}{\sqrt{\mathbf{v}_x^T \boldsymbol{\omega} \mathbf{v}_x}\sqrt{\mathbf{v'}_i^T \boldsymbol{\omega} \mathbf{v'}_i}} \tag{10}$$

Without loss of generality, we choose an arbitrary point on the horizon line as the vanishing point $\mathbf{v}_x$ along the x-axis, and the image point $\mathbf{b}$ of the footprint/bottom as the image of the world origin. The vanishing point $\mathbf{v}_y$ along the y-axis is then given by $\mathbf{v}_y \sim \boldsymbol{\omega}\mathbf{v}_x \times \boldsymbol{\omega}\mathbf{v}_z$.

Let ψ_i be the angles measured clockwise that the shadow points make with the positive x-axis as shown in Fig. 2. We have

$$\cos\psi_i = \frac{\mathbf{v'}_i^T \boldsymbol{\omega} \mathbf{v}_x}{\sqrt{\mathbf{v'}_i^T \boldsymbol{\omega} \mathbf{v'}_i}\sqrt{\mathbf{v}_x^T \boldsymbol{\omega} \mathbf{v}_x}} \tag{11}$$

$$\sin\psi_i = \frac{\mathbf{v'}_i^T \boldsymbol{\omega} \mathbf{v}_y}{\sqrt{\mathbf{v'}_i^T \boldsymbol{\omega} \mathbf{v'}_i}\sqrt{\mathbf{v}_y^T \boldsymbol{\omega} \mathbf{v}_y}} \quad i = 1, 2, 3 \tag{12}$$

Next, we define the following ratios, which are readily derived from spherical coordinates, and also used in sundial construction [1,2,3]:

$$\rho_1 = \frac{\cos\phi_2 \cos\psi_2 - \cos\phi_1 \cos\psi_1}{\sin\phi_2 - \sin\phi_1} \tag{13}$$

$$\rho_2 = \frac{\cos\phi_2 \sin\psi_2 - \cos\phi_1 \sin\psi_1}{\sin\phi_2 - \sin\phi_1} \tag{14}$$

$$\rho_3 = \frac{\cos\phi_2 \cos\psi_2 - \cos\phi_3 \cos\psi_3}{\sin\phi_2 - \sin\phi_3} \tag{15}$$

$$\rho_4 = \frac{\cos\phi_2 \sin\psi_2 - \cos\phi_3 \sin\psi_3}{\sin\phi_2 - \sin\phi_3} \tag{16}$$

For our problem, it is clear from (7)-(12) that these ratios are all determined directly in terms of image quantities. This is possible only because the camera has been calibrated. The angle measured at world origin between the positive y-axis and the ground plane's primary meridian (i.e. the north direction) is then given by

$$\alpha = \tan^{-1}\left(\frac{\rho_1 - \rho_3}{\rho_4 - \rho_2}\right) \tag{17}$$

from which we can determine the GPS latitude of the location where the pictures are taken as

$$\lambda = \tan^{-1}(\rho_1 \cos\alpha + \rho_2 \sin\alpha) \tag{18}$$

For n shadow points, we obtain a total of $\frac{n!}{(n-3)!3!}$ estimations of latitude(λ). In the presence of noise, this leads to a very robust estimation of λ.

Day Number: Once the latitude is determined from (18), we can also determine the exact day when the images are taken. For this purpose, let δ denote the declination angle (positive in the summer). Let also $\hbar$ denote the hour angle for a given image, i.e. the angle the earth needs to rotate to bring the meridian of that location to solar noon, where each hour time corresponds to $\frac{\pi}{12}$ radians, and the solar noon is when the sun is due south with maximum altitude. Then these angles are given in terms of the latitude λ, the sun's altitude ϕ and its azimuth θ by

$$\sin\hbar\cos\delta - \cos\phi\sin\theta = 0 \tag{19}$$

$$\cos\delta\cos\lambda\cos\hbar + \sin\delta\sin\lambda - \sin\phi = 0 \tag{20}$$

Again, note that the above system of equations depend only on image quantities defined in (7)-(12). Upon finding the declination and the hour angles by solving the above equations, the exact day of the year when the pictures are taken can be found by

$$N = \frac{365}{2\pi}\sin^{-1}\left(\frac{\delta}{\delta_m}\right) - N_o \tag{21}$$

where N is the day number of the date, with January 1^{st} taken as $N = 1$, and February assumed of 28 days, $\delta_m \simeq 0.408$ is the maximum absolute declination angle of earth in radians, and $N_o = 284$ corresponds to the number of days from the first equinox to January 1^{st}.

Longitude: Unfortunately, unlike latitude, the longitude cannot be determined directly from observing shadows. The longitude can only be determined either by spatial or temporal correlation. For instance, if we know that the pictures are taken in a particular state or a country or a region in the world, then we only need to perform a one-dimensional search along the latitude determined by (18) to find also the longitude and hence the GPS coordinates. Alternatively, the longitude may be determined by temporal correlation. For instance, suppose we have a few frames from a video stream of a live webcam with unknown location. Then they can be temporally correlated with our local time, in which case the difference in hour angles can be used to determine the longitude.

For this purpose, let $\hbar_l$ and γ_l be our own local hour angle and longitude at the time of receiving the live pictures. Then the GPS longitude of the location where the pictures are taken is given by

$$\gamma = \gamma_l + (\hbar - \hbar_l) \tag{22}$$

Therefore, by using only three shadow points, compared to 5 required for the camera calibration, we are able to determine the geo-location up to longitude ambiguity, and specify the day of the year when the images were taken up to, of course, year ambiguity. The key observation that allows us to achieve this is the fact that a calibrated camera performs as a direction tensor, capable of measuring direction of rays and hence angles, and that the latitude and the day of the year are determined simply by measuring angles in images.

In the next section, we validate our method and evaluate the accuracy of both self-calibration and geo-temporal localization steps using synthetic and real data.

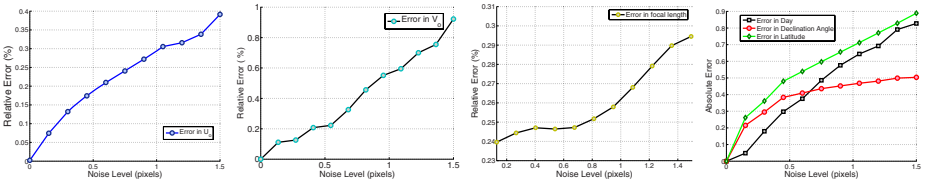
Algorithm - Geo-Temporal Localization
-Input: Shadow points of at least two objects

- Obtain the vertical vanishing point $\mathbf{v}_z$.
- Estimate the horizon line $\mathbf{l}_\infty$.
  - If the shadow casting object are visible: use the method described in Section 3.1 for $\mathbf{l}_\infty$ estimation.
  - Else, fit a conic to shadow trajectory of each object, and compute conic intersections (Section 3.2). Fit a line through the intersection points for a robust estimation of $\mathbf{l}_\infty$.
- Perform camera calibration, as described in Section 3.3.
- Estimate the altitude and the azimuth angles, eqs (7)-(10). Estimate the ratios (13)-(16) to estimate the latitude λ and N.
- If time of image acquisition is know, estimate the longitude γ.

5 Experimental Results

We rigorously tested and validated our method on synthetic as well as real data sequences for both self-calibration and geo-temporal localization steps. Results are described below.

Synthetic Data: Two vertical objects of different heights were randomly placed on the ground plane. Using the online available version of SunAngle Software [21], we generated altitude and azimuth angles for the sun corresponding to our own geo-location with



(a) (b) (c) (d)

Fig. 5. Performance averaged over 1000 independent trials: (a) & (b) relative error in the coordinates of the principal point (u_o, v_o), (c) the relative error in the focal length f. (d) Result for average error in latitude, solar declination angle, and day of the year.

Fig. 6. Few of the images taken from one of the live webcams in downtown Washington D.C. The two objects that cast shadows on the ground are shown in red and blue, respectively. Shadows move to the left of the images as time progresses.

latitude $28.51°$ *(we omit the longitude information to maintain our anonymity leaving you with one dimensional ambiguity)*. The data was generated for the 315^{th} day of the year i.e. the 11^{th} of November 2006 from `10:00am` to `2:00pm`. The solar declination angle for that time period is $-17.49°$. The vertical objects and the shadow points were projected by a synthetic camera with a focal length of $f = 1000$, the principal point at $(u_o, v_o) = (320, 240)$, unit aspect ratio, and zero skew.

In order to test resilience of the proposed self-calibration method to noise, we gradually added Gaussian noise of zero mean and standard deviation of up to 1.5 pixels to the projected points. The estimated parameters were then compared with the ground truth values mentioned above. For each noise level, we performed 1000 independent trials. The final averaged results for calibration parameters are shown in Figure 5. Note that, as explained in [15], the relative difference with respect to the focal length is a more geometrically meaningful error measure. Therefore, relative error of f, u_o and v_o were measured w.r.t f while varying the noise from 0.1 to 1.5 pixels. As shown in the figure, errors increase almost linearly with the increase of noise in the projected points. For the noise of 1.5 pixels, the error is found to be less than 0.3% for f, less than 0.5% for u_o and less than 1% for v_o.

Averaged results for latitude, solar declination angle, and the day of the year are shown in Figure 5(d). The error is found to be less than 0.9%. For a maximum noise level of 1.5 pixels, the estimated latitude is $28.21°$, the declination angle is $-17.93°$, and the day of the year is found to be 314.52.

Real Data: Several experiments on two separate data sets are reported below for demonstrating the power of the proposed method. In the first set, 11 images were captured live from downtown Washington D.C. area, using one of the webcams available online at `http://trafficland.com/`. As shown in Figure 6, a lamp post and a traffic

Table 1. Results for 11 sets of 10-image combinations. Mean value and standard deviation for latitude is found to be $(38.743°, 3.57)$, $(-16.43°, 1.11)$ for the declination angle, and $(329.95, 2.28)$ for the estimated number of the day.

	$Comb_1$	$Comb_2$	$Comb_3$	$Comb_4$	$Comb_5$	$Comb_6$	$Comb_7$	$Comb_8$	$Comb_9$	$Comb_{10}$	$Comb_{11}$
Latitude	33.73	35.70	37.03	36.1	35.72	38.21	39.23	45.78	41.84	40.88	41.96
Declination	-14.47	-15.78	-15.93	-16.54	-17.25	-16	-16.70	-18.94	-15.87	-16.99	-16.24
Day #	328.64	332.26	331.09	326.87	330.15	331.37	331.32	332.56	326.81	331.72	326.72

Fig. 7. Few of the images in the second data set that were temporally correlated with our local time, taken also from one of the live webcams in Washington D.C. The objects that cast shadows on the ground are highlighted. Shadows move to the left of the images as time progresses.

light were used as two objects casting shadows on the road. The shadow points are highlighted by colored circles in the figure. The calibration parameters were estimated as $\mathbf{K} = \begin{bmatrix} 700.36 & 0 & 172 \\ 0 & 700.36 & 124 \\ 0 & 0 & 1 \end{bmatrix}$.

Since we had more than the required minimum number of shadow locations over time, in order to make the estimation more robust to noise, we took all possible combinations of the available points and averaged the results. For this first data set the images were captured on the 15^{th} November at latitude $38.53°$ and longitude $77.02°$. We estimated the latitude as $38.74°$, the day number as 329.95 and the solar declination angle as $-16.43°$ compared to the actual day of 319, and the declination angle of $-18.62°$. The small errors can be attributed to many factors e.g. noise, non-linear distortions and errors in the extracted features in low-resolution images of 320×240. Despite all these factors, the experiment indicates that the proposed method provides good results.

In order to evaluate the uncertainty associated with our estimation, we then divided this data set into 11 sets of 10-image combinations, i.e. in each combination we left one image out. We repeated the experiment for each combination and calculated the mean and the standard deviation of the estimated unknown parameters. Results are shown in Table 5. The low standard deviations can be interpreted as small uncertainty, indicating that our method is consistently providing reliable results.

A second data set is shown in Figure 7. The ground truth for this data set was as follows: longitude $77.02°$, latitude $38.53°$, day number of 331, and the declination of $-21.8°$. For this data set we assumed that the data was downloaded in real-time and hence was temporally correlated with our local time. We estimated the longitude as $78.761°$, the latitude as $37.79°$, the day number as 323.07, and the declination angle as $-17.29°$.

6 Conclusion

We propose a method based entirely on computer vision to determine the geo-location of the camera up to longitude ambiguity, without using any GPS or other instruments, and by solely relying on imaged shadows as cues. We also describe situations where longitude ambiguity can be removed by either temporal or spatial cross-correlation. Moreover, we determine the date when the pictures are taken without using any prior information. Unlike shadow-based calibration methods such as [6,7], this step does not require the objects themselves to be seen in the images.

References

1. Herbert, A.: Sundials Old and New. Methuen & Co. Ltd. (1967)
2. III, F.W.S.: A three-point sundial construction. Bulletin of the British Sundial Society 94, 22–29 (1994)
3. Waugh, A.: Sundials: Their Theory and Construction. Dover Publications, Inc. (1973) ISBN 0-486-22947-5
4. Bouguet, J., Perona, P.: 3D photography on your desk. In: Proc. ICCV, pp. 43–50 (1998)
5. Caspi, Y., Werman, M.: Vertical parallax from moving shadows. In: Proc. CVPR, pp. 2309–2315 (2006)
6. Antone, M., Bosse, M.: Calibration of outdoor cameras from cast shadows. In: Proc. IEEE Int. Conf. Systems, Man and Cybernetics, pp. 3040–3045 (2004)
7. Cao, X., Shah, M.: Camera calibration and light source estimation from images with shadows. In: Proc. IEEE CVPR, pp. 918–923 (2005)
8. Cao, X., Foroosh, H.: Camera calibration and light source orientation from solar shadows. Journal of Computer Vision and Image Understanding (CVIU) 105, 60–72 (2006)
9. Lu, F., Shen, Y., Cao, X., Foroosh, H.: Camera calibration from two shadow trajectories. In: Proc. ICPR, pp. 1–4 (2005)
10. Heikkila, J.: Geometric camera calibration using circular control points. IEEE Trans. Pattern Anal. Mach. Intell. 22, 1066–1077 (2000)
11. Hartley, R.I.: Self-calibration of stationary cameras. Int. J. Comput. Vision 22, 5–23 (1997)
12. Heyden, A., Astrom, K.: Euclidean reconstruction from image sequences with varying and unknown focal length and principal point. In: Proc. IEEE CVPR, pp. 438–443 (1997)
13. Pollefeys, M., Koch, R., Gool, L.V.: Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. Int. J. Comput. Vision 32, 7–25 (1999)
14. Liebowitz, D., Zisserman, A.: Combining scene and auto-calibration constraints. In: Proc. IEEE ICCV, pp. 293–300 (1999)
15. Triggs, B.: Autocalibration from planar scenes. In: Proc. ECCV, pp. 89–105 (1998)
16. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. 22, 1330–1334 (2000)
17. Jacobs, N., Satkin, S., Roman, N., Speyer, R., Pless, R.: Geolocating static cameras. In: Proc. of ICCV, pp. 469–476 (2007)
18. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 25, 564–575 (2003)
19. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
20. Junejo, I., Foroosh, H.: Dissecting the image of the absolute conic. In: 5th IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS) (2006)
21. Gronbeck, C.: Sunangle software, `www.susdesign.com/sunangle/`
22. Semple, J.G., Kneebone, G.T.: Algebraic Projective Geometry. Oxford Classic Texts in the Physical Sciences (1979)

APPENDIX

A Computing Conic Intersection

Two conics always intersect at four points, which may be real or imaginary. All conics passing through the four points of intersection can be written as

$$\mathbf{C}_\mu \sim \mathbf{C}_1 + \mu \mathbf{C}_2. \tag{A-1}$$

Equation (A-1) defines a pencil of conics parameterized by μ, where all the conics in the pencil intersect at the same four points $\mathbf{m}_i, i = 1, ..., 4$. Four such points such that no three of them are collinear also give rise to what is known as the *complete quadrangle*.

It can be shown that in this pencil at most three conics are not full rank. For this purpose note that any such degenerate conic should satisfy

$$\det(\mathbf{C}_\mu) = \det(\mathbf{C}_1 + \mu \mathbf{C}_2) = 0. \tag{A-2}$$

It can then be readily verified that (A-2) is a cubic equation in terms of μ. Therefore upon solving (A-2), we obtain at most three distinct values $\mu_i, i = 1, ..., 3$, which provide the three corresponding degenerate conics

$$\mathbf{C}_{\mu_i} \sim \mathbf{C}_1 + \mu_i \mathbf{C}_2, \quad i = 1, ..., 3. \tag{A-3}$$

In the general case (i.e. when the three parameters $\mu_i, i = 1, ..., 3$ are distinct), the three degenerate conics are of rank 2, and therefore can be written as

$$\mathbf{C}_{\mu_i} \sim \mathbf{l}_i \mathbf{l'}_i^T + \mathbf{l}'_i \mathbf{l}_i^T, \quad i = 1, ..., 3, \tag{A-4}$$

where $\mathbf{l}_i$ and $\mathbf{l}'_i$ are three pairs of lines as shown in Fig.3.

Now, let $\mathbf{C}^*_{\mu_i}$ be the adjoint matrix of $\mathbf{C}_{\mu_i}$. It then follows from (A-4) that

$$\mathbf{C}^*_{\mu_i} \mathbf{l}_i = \mathbf{C}^*_{\mu_i} \mathbf{l}'_i = 0, \quad i = 1, ..., 3, \tag{A-5}$$

which yields (by using the property that the cofactor matrix is related to the way matrices distribute with respect to the cross product [19])

$$\mathbf{C}^*_{\mu_i} \mathbf{l}_i \times \mathbf{C}^*_{\mu_i} \mathbf{l}'_i = \mathbf{C}_{\mu_i} (\mathbf{l}_i \times \mathbf{l}'_i) = 0, \quad i = 1, ..., 3. \tag{A-6}$$

In other words, the intersection point $\mathbf{v}_i$ of the pair of lines, $\mathbf{l}_i$ and $\mathbf{l}'_i$, is given by the right null space of $\mathbf{C}_{\mu_i}$. Therefore, in practice, it can be found as the eigenvector corresponding to the smallest eigenvalue of the degenerate conic $\mathbf{C}_{\mu_i}$. The triangle formed by the three vertices $\mathbf{v}_1 \mathbf{v}_2$ and $\mathbf{v}_3$ is known as the *diagonal triangle* of the quadrangle [22].

Next, we verify that for any conic $\mathbf{C}_\mu$ in the pencil

$$(\mathbf{l}_i \times \mathbf{l}'_i)^T \mathbf{C}_\mu (\mathbf{l}_j \times \mathbf{l}'_j) = 0, \quad i \neq j, \quad i, j = 1, ..., 3 \tag{A-7}$$

This means that any pair of right null vectors of the degenerate conics $\mathbf{C}_{\mu_i}, i = 1, ..., 3$ are conjugate with respect to all conics in the pencil. In other words, their intersections form the vertices of a self-polar triangle with respect to all the conics in the pencil.

To obtain the intersection points of the two shadow conics, we use the fact that all the conics in the pencil intersect at the same four points. Therefore, the intersection points can also be found as the intersection of the lines $\mathbf{l}_i$ and $\mathbf{l}'_i$ with the lines $\mathbf{l}_j$ and $\mathbf{l}'_j$ $(i \neq j)$. The lines $\mathbf{l}_i$ and $\mathbf{l}'_i$ can be simply found by solving

$$\mathbf{C}_{\mu_i} \sim \mathbf{l}_i \mathbf{l'}_i^T + \mathbf{l}'_i \mathbf{l}_i^T \tag{A-8}$$

Equation (A-8) provides 4 constraints on $\mathbf{l}_i$ and $\mathbf{l}'_i$ (5 due to symmetry minus 1 for rank deficiency). In practice it leads to two quadratic equations on the four parameters of the two lines, which can be readily solved. The solution, of course, has a twofold ambiguity due to the quadratic orders, which is readily resolved by the fact that

$$\mathbf{l}_i \times \mathbf{l}'_i \sim \text{null}(\mathbf{C}_{\mu_i}) \tag{A-9}$$

The process can be repeated for $\mathbf{l}_j$ and $\mathbf{l}'_j$, and the intersections of the lines between the two sets would then provide the four intersection points of the shadow conics.

An Experimental Comparison of Discrete and Continuous Shape Optimization Methods

Maria Klodt, Thomas Schoenemann, Kalin Kolev,
Marek Schikora, and Daniel Cremers

Department of Computer Science,
University of Bonn, Germany
{klodt,tosch,kolev,schikora,dcremers}@cs.uni-bonn.de

Abstract. Shape optimization is a problem which arises in numerous computer vision problems such as image segmentation and multiview reconstruction. In this paper, we focus on a certain class of binary labeling problems which can be globally optimized both in a spatially discrete setting and in a spatially continuous setting. The main contribution of this paper is to present a quantitative comparison of the reconstruction accuracy and computation times which allows to assess some of the strengths and limitations of both approaches. We also present a novel method to approximate length regularity in a graph cut based framework: Instead of using pairwise terms we introduce higher order terms. These allow to represent a more accurate discretization of the L_2-norm in the length term.

1 Introduction

Shape optimization is at the heart of several classical computer vision problems. Following a series of seminal papers [2, 12, 15, 21, 27], functional minimization has become the established paradigm for these problems. In the spatially discrete setting the study of the corresponding binary labeling problems goes back to the spin-glas models introduced in the 1920's [19]. In this paper, we focus on a class of functionals of the form:

$$E(S) = \int_{int(S)} f(x)\,\mathrm{d}^n x \;+\; \nu \int_S g(x)\,\mathrm{d}S, \tag{1}$$

where S denotes a hypersurface in $\mathbb{R}^n$, i.e. a set of closed boundaries in the case of $2D$ image segmentation or a set of closed surfaces in the case of $3D$ segmentation and multiview reconstruction. The functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^+$ are application dependent. In a statistical framework for image segmentation, for example, $f(x) = \log p_{bg}(I(x)) - \log p_{ob}(I(x))$ may denote the log likelihood ratio for observing the intensity $I(x)$ at any given point x given that x is part of the background or the object, respectively.

The second term in (1) corresponds to an isotropic measure of area (for $n = 3$) or boundary length ($n = 2$), measured by the function g.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 332–345, 2008.

Input image | Intensity-based segmentation | One of the input images | Reconstructed object

Fig. 1. Examples of shape optimization: Image segmentation and 3D reconstruction

In the context of image segmentation, g may be a measure of the local edge strength – as in the geodesic active contours [6, 22] – which energetically favors segmentation boundaries along strong intensity gradients. In the context of multiview reconstruction, $g(x)$ is typically a measure of the consistency among different views of the voxel x, where low values of g indicate a strong agreement from different cameras on the observed patch intensity – see for example [12]. Figure 1 shows examples of shape optimization using the example of image segmentation and multiview reconstruction.

Functionals of the form (1) can be globally optimized by reverting to implicit representations of the hypersurface S using an indicator function $u : \mathbb{R}^n \rightarrow \{0, 1\}$, where $u = 1$ and $u = 0$ denote the interior and exterior of S. The functional (1) defined on the space of surfaces S is therefore equivalent to the functional

$$E(u) = \int_{\mathbb{R}^n} f(x)\, u(x)\, \mathrm{d}^n x + \nu \int_{\mathbb{R}^n} g(x)\, |\nabla u(x)|\, \mathrm{d}^n x, \tag{2}$$

defined on the space of binary labelings u, where the second term in (2) is the weighted total variation norm which can be extended to non-differentiable functions in a weak sense.

In the current experimental comparison, we focus on functionals of the type (1) since they allow for the efficient computation of globally optimal solutions of region-based functionals. There exist numerous alternative functionals for shape optimization, including ratio functionals [20, 29]. Recently it was shown that some region-based ratio functionals can be optimized globally [25]. As this method did not yet reach a high popularity, we leave it for future discussion.

The functional (2) can be globally optimized in a spatially discrete setting: By mapping each labeling to a cut in a graph, the problem is reduced to computing the minimal cut. First suggested in [18], it was later rediscovered in [5] and has since become a popular framework for image segmentation [28] and multiview reconstruction [31]. More recently it was shown in [7, 8] that the same binary labeling problem (2) can be globally minimized in a spatially *continuous* setting as well. An alternative spatially continuous formulation of graph cuts was developed in [1].

In this paper, we propose the first quantitative experimental comparison of spatially discrete and spatially continuous optimization methods for functionals

of the form (2). In particular, we focus on the quality and efficiency of shape optimization in discrete and continuous setting. Furthermore we propose a new approximation of the L_2-norm in the context of graph cuts based optimization.

2 Spatially Discrete Optimization Via Graph Cuts

To solve the binary labeling problem (2) in a discrete setting, the input data is converted into a directed graph in form of a regular lattice: Each pixel (or voxel) in the input data corresponds to a node in the lattice. To approximate the metric g measuring the boundary size of the hypersurface S, neighboring nodes are connected. The degree of connectivity depends on the application. We defer details to Section 2.2.

Additionally a source node s and a sink node t are introduced. They allow to include the unary terms $f(x)u(x)$ for the pixels x: If $f(x) \geq 0$, an edge to the source is introduced, weighted with $f(x)$. Otherwise an edge to the sink weighted with $-f(x)$ is created.

The optimal binary labeling u corresponds to the minimal s/t-cut in the graph. An s/t-cut is a partitioning of the nodes in the graph into two sets S and T, where S contains the source s and T the sink t. Nodes $x \in S$ are assigned the label $u(x) = 0$, nodes $x \in T$ the label $u(x) = 1$. The weight of such a cut is the sum of the weights of all edges starting in S and ending in T.

2.1 Computing the Minimal Cut in a Graph

Efficient solvers of the minimal s/t-cut problem are based on computing the maximal flow in the graph [13]. Such methods are divided into three major categories: those based on augmenting paths [4, 11, 13], blocking flows [10, 17] and the push-relabel method [16]. Some of these methods do not guarantee a polynomial running time [4] or require integral edge weights [17]. To solve 2-dimensional problems of form (2) usually the algorithm of Boykov and Kolmogorov performs best [4]. For highly connected three-dimensional grids the performance of this algorithm breaks down [4] and push-relabel methods become competitive. Recently efforts were made to parallelize push-relabel-based approaches [9].

2.2 Approximating Metrics Using Graph Cuts

The question of how to approximate continuous metrics of the boundary size in a discrete setting has received significant attention by researchers. Boykov and Kolmogorov [3] show how to approximate any Riemannian metric, including anisotropic ones. In [24] they discuss how to integrate flux. A similar construction can be derived from the divergence theorem. In the following we limit our discussion to the isotropic case.

We start with a review of the method in [3] which replaces the L_2-norm of the gradient in (2) by its L_1-norm. For the Euclidean metric ($g(x) = 1 \; \forall x \in \mathbb{R}^n$) we then propose a novel discretization scheme which allows to use the L_2-norm of the gradient by introducing higher order terms.

Approximation Using Pairwise Terms. Based on the Cauchy-Crofton formula of integral geometry, Boykov and Kolmogorov [3] showed that the metric given by g can be approximated by connecting pixels to all pixels in a given neighborhood. The respective neighborhood systems can be expressed as

$$N_R(x) = \left\{ x + \begin{pmatrix} a \\ b \end{pmatrix} \middle| a, b \in \mathbb{Z},\ \sqrt{a^2 + b^2} \leq R,\ gcd(|a|, |b|) = 1 \right\}.$$

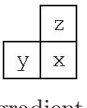
The constraint on the greatest common divisor avoids duplicate directions. The edge corresponding to $(a\ b)^\top$ is given a weight of $g(x)/\sqrt{a^2 + b^2}$. For $R = 1$ the obtained 4-connected lattice reflects the L_1-norm of the gradient. With increasing R and decreasing grid spacing the measure converges to the continuous measure. This is not true when fixing the connectivity (i.e. when keeping R constant).

A Novel Length Approximation Using Higher Order Terms. The energy (2) involves the L_2-norm of the generalized gradient of the $\{0, 1\}$-function u. With the pairwise terms discussed above a large connectivity is needed to approximate this norm. In the following, we will show that a more accurate approximation of the L_2-norm can be integrated in a graph cut framework, without increasing the connectivity. The key observation is that in a two-dimensional space a consistent calculation of the gradient is obtained by taking the differences to the upper and left neighbor in the grid - see Figure 2.

The Figure also shows the arising term. One easily verifies that this term satisfies the submodularity condition [26]. For a third order term as this one, this condition implies that the term can be minimized using graph cuts.

We also considered the corresponding term in 3D space where each pixel is connected to three neighbors. The arising fourth order term – with values in $\{0, 1, \sqrt{2}, \sqrt{3}\}$ – is submodular. However it is not clear whether it can be minimized via graph cuts: It does not satisfy the sufficient conditions pointed out by Freedman [14].

From a practical point of view, in 2D the novel terms do not perfom well: The length discretization only compares a pixel to those pixels in the direction of the upper left quadrant. Performance is boosted when adding the respective terms for the other three quadrants as well.



gradient mask

$u(x)$	$u(y)$	$u(z)$	$\lvert\nabla u\rvert$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	$\sqrt{2}$

$u(x)$	$u(y)$	$u(z)$	$\lvert\nabla u\rvert$
1	0	0	$\sqrt{2}$
1	0	1	1
1	1	0	1
1	1	1	0

Fig. 2. The L_2-norm of the 2D gradient as a ternary term. One easily verifies that this term is submodular.

3 Spatially Continuous Optimization Via Relaxation

More recently, it was shown that the class of functionals (2) can also be minimized in a spatially continuous setting by reverting to convex relaxations [7, 8]. By relaxing the binary constraint and allowing the function u to take on values in the interval between 0 and 1, the optimization problem becomes minimizing the convex functional (2) over the convex set

$$u : \mathbb{R}^n \rightarrow [0, 1]. \tag{3}$$

Global minimizers u^* of this relaxed problem can efficiently be computed (see section 3.2).

3.1 Convex Relaxation and the Thresholding Theorem

The following theorem [8, 30] assures that thresholding the solution u^* of the relaxed problem provides a minimizer of the original binary labeling problem (2). In other words the convex relaxation preserves global optimality for the original binary labeling problem.

Theorem 1. *Let $u^* : \mathbb{R}^n \rightarrow [0, 1]$ be a global minimizer of the functional (2). Then all upper level sets (i.e. thresholded versions)*

$$\Sigma_{\mu,u^*} = \{x \in \mathbb{R}^n \,|\, u^*(x) > \mu\}, \qquad \mu \in (0, 1), \tag{4}$$

of u^* are minimizers of the original binary labeling problem (1).

Proof. Using the layer cake representation of the function $u^* : \mathbb{R}^n \rightarrow [0, 1]$:

$$u^*(x) = \int_0^1 1_{\Sigma_{\mu,u*}}(x)\, \mathrm{d}\mu \tag{5}$$

we can rewrite the first term in the functional (2) as

$$\int_{\mathbb{R}^n} f u^* \,\mathrm{d}x = \int_{\mathbb{R}^n} f \left(\int_0^1 1_{\Sigma_{\mu,x}} \,\mathrm{d}\mu \right) \mathrm{d}x = \int_0^1 \int_{\Sigma_{\mu,u^*}} f(x) \,\mathrm{d}x \tag{6}$$

As a consequence, the functional (2) takes on the form:

$$E(u^*) = \int_0^1 \Big\{ \int_{\Sigma_{\mu,u^*}} f \,\mathrm{d}x \,+\, \big|\partial\Sigma_{\mu,u^*}\big|_g \Big\} \,\mathrm{d}\mu \;\equiv\; \int_0^1 \hat{E}\big(\Sigma_{\mu,u^*}\big) \,\mathrm{d}\mu, \tag{7}$$

where we have used the coarea formula to express the weighted total variation norm in (2) as the integral over the length of all level lines of u measured in the norm induced by g. Clearly the functional (7) is now merely an integral of the original binary labeling problem $\hat{E}$ applied to the upper level sets of u^*.

Assume that for some threshold value $\tilde{\mu} \in (0, 1)$ theorem 1 was not true, i.e. there exists a minimizer Σ^* of the binary labeling problem with smaller energy:

$$\hat{E}(\Sigma^*) < \hat{E}(\Sigma_{\tilde{\mu},u^*}). \tag{8}$$

Then for the indicator function 1_{Σ^*} of the set Σ^* we have:

$$E(1_{\Sigma^*}) = \int_0^1 \hat{E}(\Sigma^*)\,\mathrm{d}\mu < \int_0^1 \hat{E}(\Sigma_{\mu,u^*})\,\mathrm{d}\mu = E(u^*), \tag{9}$$

which contradicts the assumption that u^* was a global minimizer of (2). □

Global minimizers of the functional (2) in a spatially continuous setting are therefore calculated as follows:

1. Compute a minimizer u^* of the energy (2) on the convex set of functions $u : \mathbb{R}^n \to \mathbb{R}$. Details are given in section 3.2.
2. Threshold the minimizer u^* at some value $\mu \in (0, 1)$ to obtain a binary solution of the original shape optimization problem. Although these solutions generally depend on μ, all of them are guaranteed to be global minimizers of (2). In all experiments in this paper we set $\mu = 0.5$.

3.2 Numerical Implementation

A minimizer of (2) must satisfy the Euler-Lagrange equation

$$0 = f(x) - \nu \operatorname{div}\left(g(x)\frac{\nabla u(x)}{|\nabla u(x)|}\right) \qquad \forall x \in \mathbb{R}^n. \tag{10}$$

Solutions to this system of equations can be obtained by a large variety of numerical solvers. We discuss some of them in the following.

Gradient Descent. The right hand side of (10) is the functional derivative of the energy (2) and gives rise to a gradient descent scheme. In practice such schemes are known to converge very slowly.

Linearized Fixed-Point Iteration. Discretization of the Euler-Lagrange equation (10) leads to a sparse nonlinear system of equations. This can be solved using a fixed point iteration scheme that transforms the nonlinear system into a sequence of linear systems. These can be efficiently solved with iterative solvers, such as Jacobi, Gauss-Seidel, Successive over-relaxation (SOR), or even multigrid methods (also called FAS for "full approximation schemes").

The only source of nonlinearity in (10) is the diffusivity $d := \frac{g}{|\nabla u|}$. Starting with an (arbitrary) initialization, one alternates computing the diffusivities and solving the *linear* system of equations with fixed diffusivities. We choose the SOR method as in [23].

Parallelization on Graphics Processing Unit. PDE-based approaches are generally suitable for parallel computing on graphics cards: The gradient descent and Jacobi schemes are straightforward to parallelize. This does not hold for the standard Gauss-Seidel scheme as it requires sequential processing of the image. However, in its Red-Black variant the Gauss Seidel scheme is parallelizable. The same holds for its various derivates such as SOR and FAS.

4 Quantitative Comparison

This section constitutes the main contribution of this paper. It provides a detailed quantitative comparison of the spatially discrete and spatially continuous shape optimization schemes introduced above. While both approaches aim at minimizing the same functional, we identified three important differences:

- The spatially discrete approach has an exact termination criterion and a guaranteed polynomial running time (for a number of maximum-flow algorithms). On the other hand, the spatially continuous approach is based on the iterative minimization of a non-linear convex functional. While the required number of iterations is typically size-independent (leading to a computation time which is linear in the number of voxels), one cannot speak of a guaranteed polynomial time complexity.
- The spatially discrete approach is based on discretizing the cost functional on a lattice and minimizing the resulting submodular problem by means of graph cuts. The spatially continuous approach, on the other hand is based on minimizing the relaxed problem in a continuous setting where the resulting Euler-Lagrange equations are solved on a discrete lattice. This difference gives rise to metrication errors of the spatially discrete approach which will be discussed in Section 4.1.
- The optimization of the spatially discrete approach is based on solving a maximum flow problem, whereas the spatially continuous approach is performed by solving a partial differential equation. This fundamental difference in the underlying computational machinery leads to differences in computation time, memory consumption and parallelization properties.

4.1 Metrication Errors and Consistency

Figure 4 shows a comparison of graph cut approaches with the continuous total variation (TV) segmentation, where we show several ways to deal with the discretization of the metric for graph cuts. None of the graph cut approaches produces such a smooth curve as the TV segmentation, although the 16-connected grid gets quite close to it. This inspired us to investigate the source for the metrication errors arising in graph cut methods.

On the 4-connected grid in $\mathbb{R}^2$, for example, graph cuts usually approximate the Euclidean boundary length of the interface S as

$$|S| = \int_S dS \approx \frac{1}{2} \sum_i \sum_{j \in \mathcal{N}(i)} |u_i - u_j|, \tag{11}$$

where $\mathcal{N}(i)$ denotes the four neighbors of pixel i. This implies that the boundary length is measured in an L_1-norm rather than the L_2-norm corresponding to the Euclidean length. The L_1 norm clearly depends on the choice of the underlying grid and is not rotationally invariant. Points of constant distance in this norm form a diamond rather than a circle (see Figure 3). This leads to a preference of boundaries along the axes (see fig. 4(a)).

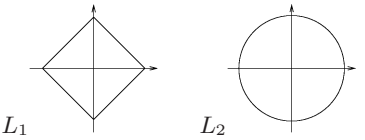


Fig. 3. 2D visualization of the L_1-norm and the L_2-norm for points of constant distance: Unlike the L_1-norm, the L_2-norm is rotationally invariant

This dependency on the underlying grid can be reduced by increasing the neighborhood connectivity. By reverting to larger and larger neighborhoods one can gradually eliminate the metrication error [3]. Increasing the connectivity leads in fact to better and better approximations of the Euclidean L_2-norm (see fig. 4(b) and 4(c)).

Yet, a computationally efficient solution to the labeling problem requires to fix a choice of connectivity. And for any such choice, one can show that the metrication error persists, that the numerical scheme is not *consistent* in the sense that a certain residual reconstruction error (with respect to the ground truth) remains and cannot be eliminated by increasing the resolution.

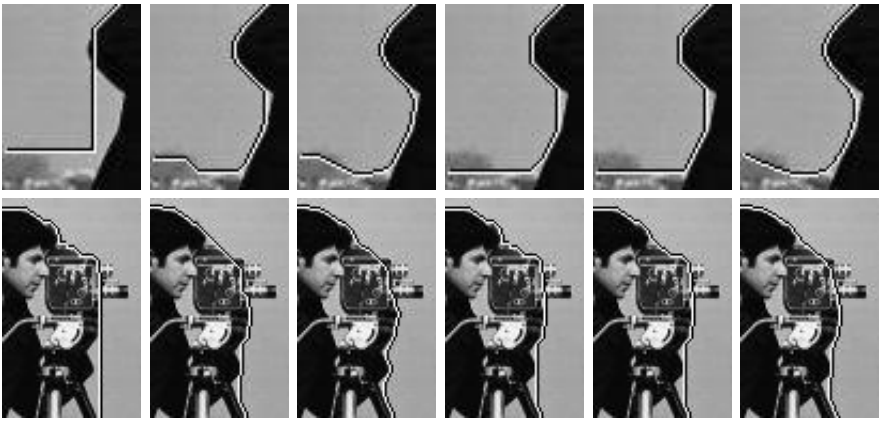
Since the spatially continuous formulation is based on a representation of the boundary length by the L_2-norm:

$$|S| = \int_S \mathrm{d}S = \int |\nabla u| \, \mathrm{d}x = \int \sqrt{u_x^2 + u_y^2} \, \mathrm{d}x, \tag{12}$$

the resulting continuous numerical scheme does not exhibit such metrication errors (see fig. 4(f)). The TV segmentation performs optimization in the convex set of functions with range in $[0, 1]$. It hence allows intermediate values where the graph cut only allows binary values.

The proposed third order graph cuts discretization of the L_2-norm (see fig. 4(d) and 4(e)) computes the same discretization of the L_2-norm, however allowing only for binary values. Hence, in this discretized version, the Euclidean length is computed for angles of 45° and 90° to the grid, by using only a 4-connected grid. Therefore the third order L_2-norm leads to similar results on a 4-connected grid as second order terms on an 8-connected grid.

Figure 5 shows a synthetic experiment of solving a minimal surface problem with given boundary constraints using the example of a bounded catenoid. As the true solution of this problem can be computed analytically, it is suitable for a comparison of different solvers. The experiment compares graph cuts and continuous TV minimization. It demonstrates that the 6-neighborhood graph cuts method completely fails to reconstruct the correct surface topology – in contrast to the full 26-neighborhood which approximates the Euclidean metric in a better way. However, discretization artifacts are still visible in terms of polyhedral blocky structures. Figure 5 also shows the deviation of the computed catenoid solutions from the analytic ground-truth for increasing volume resolution. It shows that for a fixed connectivity structure the computed graph cut



(a) 4-conn. Graph cuts (L_1) (b) 8-conn. Graph cuts (c) 16-conn. Graph cuts (d) 4-conn. Graph cuts (third order) (e) 4-conn. Graph cuts (third order) (symmetric) (f) TV segmentation (cont. L_2)

Fig. 4. Comparison of different norms and neighborhood connectivities for discrete and continuous optimization for image segmentation (Close ups of the cameraman image from figure 1). The experiment shows that a 16-connected graph is needed for the discrete solution to obtain similiar results to the continuous solution.

solution is not consistent with respect to the volume resolution. In contrast, for the solution of the continuous TV minimization the discretization error decays to zero.

Figure 6 shows an experiment for real image data. In this multiview reconstruction problem the data fidelity term is dominant, therefore the discrete and the continuous solutions are similar for the same volume resolution ($108 \times 144 \times 162$). Increasing the volume resolution to $216 \times 288 \times 324$ gives more accurate results for the continuous TV formulation, while a graph cut solution for this resolution was not feasible to compute because of RAM overflow.

4.2 Computation Time

Numerous methods exist to solve either the discrete or the continuous optimization tasks. A comparison of all these methods is outside the scope of our paper. Instead we pick a few solvers we consider competetive. For all graph cut methods we use the algorithm in [4], which is arguably the most frequently used in Computer Vision. We test all discretizations mentioned above.

For the TV segmentation we implemented sequential methods on the CPU and parallel solvers on a Geforce GTX 8800 graphics card using the CUDA framework. Both implementations are based on the SOR method. On the CPU we use the usual sequential order of pixels, and on the GPU the corresponding parallelizable Red-Black scheme. A termination criterion is necessary as the number of required iterations depends on the length weight ν. We compare the

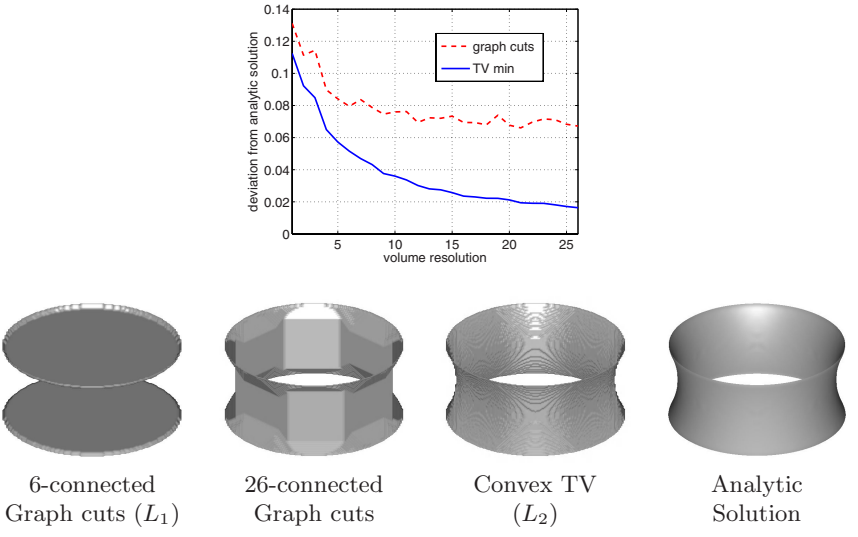


Fig. 5. Comparison of discrete and continuous optimization methods for the reconstruction of a catenoid: While the discrete graph cut algorithm exhibits prominent metrication errors (polyhedral structures), the continuous method does not show these. The plot shows the accuracy of the 26-connected graph cuts and the continuous TV method in dependence of the volume resolution. The consistency of the continuous solution is validated experimentically in the sense that the reconstruction error goes to zero with increasing resolution.

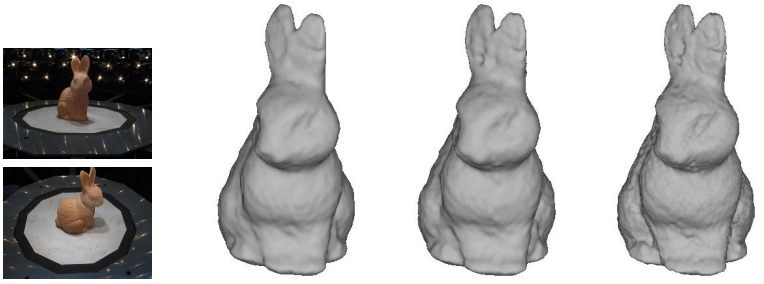
segmentations every 50 iterations and stop as soon as the maximal absolute difference drops below a value of 0.000125.

Evaluation for 2D Shape Optimization. Table 1 shows run-times for all mentioned methods. The task is image segmentation using a piecewise constant Mumford-Shah with fixed mean values 0 and 1. The main conclusions are summarized as follows:

- The TV segmentation profits significantly from parallel architectures. According to our results this is roughly a factor of 5. It should be noted that the GPU-implementation usually requires more iterations as the Red-Black order is used.
- The graph cut based methods clearly outperform the TV segmentation.
- While for the graph cut methods the 16-connected pairwise terms give generally the best results (they are largely free from grid bias), they also use up the most run-time.

Evaluation for 3D shape optimization. Table 2 shows run-times of the different optimization methods for the 3D catenoid example shown in figure 5. We detect three main conclusions:

- The 6-connected graph cuts method is the fastest, however it computes the wrong solution (see figure 5).



Two of 33 input images | Graph Cuts (108×144×162) | Convex TV (108×144×162) | Convex TV (216×288×324)

Fig. 6. Comparison of discrete and continuous optimization for multiview 3D reconstruction (presented in [23]): Due to the dominant data fidelity term, the discrete and continuous reconstructions are similar for the same volume resolution. However, for increasing resolution more accurate results can be achieved with the continuous formulation, while graph cuts rapidly come across memory limitations.

Table 1. 2D image segmentation: Run-times for the different optimization methods on two different images

	Cameraman Image			Berkeley Arc Image		
Method	$\nu = 1$	$\nu = 3$	$\nu = 5$	$\nu = 1$	$\nu = 3$	$\nu = 5$
Graph Cuts 4-connected	0.02s	0.1s	0.33s	0.06s	0.16s	0.53s
Graph Cuts 8-connected	0.05s	0.15s	0.4s	0.1s	0.27s	0.93s
Graph Cuts 16-connected	0.2s	0.35s	0.95s	0.33s	0.85s	2.7s
Graph Cuts L2 (1 quadrant)	0.03s	0.15s	0.45s	0.06s	0.19s	0.8s
Graph Cuts L2 (4 quadrants)	0.1s	0.25s	0.86	0.23s	0.53s	1.8s
TV w/ gradient descent (CPU)	111.38s	251.97s	259.87s	409.08s	636.28s	157.64s
TV w/ SOR (CPU)	10.9s	13.26s	10.2s	35.89s	103.5s	39.26s
TV w/ red-black SOR (GPU)	2s	2.7s	2s	7.6s	28.3s	8.6s

Table 2. Run-times for the 3D catenoid example

Graph cuts 6-connected	13 s
Graph cuts 26-connected	12 min 35 s
TV w/ SOR (CPU)	9 min 36 s
TV w/ red-black SOR (GPU)	30 s

– The run-time of the graph cut method changes for the worse with high connectivities, and gets slower than the TV optimization, both on CPU and GPU. Note that this limitation is due to the fact that the Boykov-Kolmogorov algorithm [4] is optimized for sparse graph structures. For denser (3D) graphs alternative push-relabel algorithms might be faster.
– The parallel implementation of the TV method allows for a speed up factor of about 20 compared to the CPU version.

4.3 Memory Consumption

With respect to the memory consumption the TV segmentation is the clear winner: It requires only one floating point value for each pixel in the image. In contrast, graph cut methods require an explicit storage of edges as well as one flow value for each edge. This difference becomes important for high resolutions, as can be seen in the experiment in figure 6.

5 Conclusion

A certain class of shape optimization functionals can be globally minimized both in a spatially discrete and in a spatially continuous setting. In this paper, we reviewed these recent developments and presented an experimental comparison of the two approaches regarding the accuracy of reconstructed shapes and computational speed. A detailed quantitative analysis confirms the following differences:

- Spatially discrete approaches generally suffer from metrication errors in the approximation of geometric quantities such as boundary length or surface area. These arise due to the binary optimization on a discrete lattice. These errors can be alleviated by reverting to larger connectivity. Alternatively, we showed that higher-order terms allow to implement an L_2-norm of the gradient, thereby providing better spatial consistency without extending the neighborhood connectivity. As the spatially continuous formulation is not based on a discretization of the cost functional but rather a discretization of the numerical optimization (using real-valued variables), it does not exhibit metrication errors in the sense that the reconstruction errors decay to zero as the resolution is increased.
- The spatially continuous formulation allows for a straight-forward parallelization of the partial differential equation. As a consequence, one may obtain lower computation times than respective graph cut methods, in particular for the denser graph structures prevalent in 3D shape optimization.
- While the discrete graph cut optimization can be performed in guaranteed polynomial time, this is not the case for the analogous continuous shape optimization. While respective termination criteria for the convex optimization work well in practice, defining termination criteria that apply to any shape optimization problem remains an open problem.

Acknowledgments

This research was supported by the German Research Foundation, grant #CR 250/3-1.

References

1. Appleton, B., Talbot, H.: Globally minimal surfaces by continuous maximal flows. IEEE Trans. Pattern Anal. Mach. Intell. 28(1), 106–118 (2006)
2. Blake, A., Zisserman, A.: Visual Reconstruction. MIT Press, Cambridge (1987)

3. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: IEEE Int. Conf. on Computer Vision, Nice, pp. 26–33 (2003)
4. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. on Patt. Anal. and Mach. Intell. 26(9), 1124–1137 (2004)
5. Boykov, Y., Veksler, O., Zabih, R.: Markov random fields with efficient approximations. In: Proc. IEEE Conf. on Comp. Vision Patt. Recog. (CVPR 1998), Santa Barbara, California, pp. 648–655 (1998)
6. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. In: Proc. IEEE Intl. Conf. on Comp. Vis., Boston, USA, pp. 694–699 (1995)
7. Chambolle, A.: Total variation minimization and a class of binary MRF models. In: Rangarajan, A., Vemuri, B.C., Yuille, A.L. (eds.) EMMCVPR 2005. LNCS, vol. 3757, pp. 136–152. Springer, Heidelberg (2005)
8. Chan, T., Esedoḡlu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. SIAM Journal on Applied Mathematics 66(5), 1632–1648 (2006)
9. Delong, A., Boykov, Y.: A scalable graph-cut algorithm for n-d grids. In: Int. Conf. on Computer Vision and Pattern Recognition, Anchorage, Alaska (2008)
10. Dinic, E.A.: Algorithm for the solution of a problem of maximum flow in a network with power estimation. Soviet Mathematics Doklady 11, 1277–1280 (1970)
11. Edmonds, J., Karp, R.: Theoretical improvements in algorithmic efficiency for network flow problems. Journal of the ACM 19, 248–264 (1972)
12. Faugeras, O., Keriven, R.: Variational principles, surface evolution, PDE's, level set methods, and the stereo problem. IEEE Trans. on Image Processing 7(3), 336–344 (1998)
13. Ford, L., Fulkerson, D.: Flows in Networks. Princeton University Press, Princeton (1962)
14. Freedman, D., Drineas, P.: Energy minimization via graph cuts: settling what is possible. In: Int. Conf. on Computer Vision and Pattern Recognition, San Diego, USA, vol. 2, pp. 939–946 (June 2005)
15. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. on Patt. Anal. and Mach. Intell. 6(6), 721–741 (1984)
16. Goldberg, A., Tarjan, R.: A new approach to the maximum flow problem. Journal of the ACM 35(4), 921–940 (1988)
17. Goldberg, A.V., Rao, S.: Beyond the flow decomposition barrier. Journal of the ACM 45, 783–797 (1998)
18. Greig, D.M., Porteous, B.T., Seheult, A.H.: Exact maximum *a posteriori* estimation for binary images. J. Roy. Statist. Soc., Ser. B 51(2), 271–279 (1989)
19. Ising, E.: Beitrag zur Theorie des Ferromagnetismus. Zeitschrift für Physik 23, 253–258 (1925)
20. Jermyn, I.H., Ishikawa, H.: Globally optimal regions and boundaries as minimum ratio weight cycles. IEEE Trans. on Patt. Anal. and Mach. Intell. 23(10), 1075–1088 (2001)
21. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Int. J. of Computer Vision 1(4), 321–331 (1988)
22. Kichenassamy, S., Kumar, A., Olver, P.J., Tannenbaum, A., Yezzi, A.J.: Gradient flows and geometric active contour models. In: IEEE Int. Conf. on Computer Vision, pp. 810–815 (1995)

23. Kolev, K., Klodt, M., Brox, T., Esedoglu, S., Cremers, D.: Continuous global optimization in multiview 3d reconstruction. In: Yuille, A.L., Zhu, S.-C., Cremers, D., Wang, Y. (eds.) EMMCVPR 2007. LNCS, vol. 4679, pp. 441–452. Springer, Heidelberg (2007)
24. Kolmogorov, V., Boykov, Y.: What metrics can be approximated by Geo Cuts or global optimization of length/area and flux. In: IEEE Int. Conf. on Computer Vision, Beijing (2005)
25. Kolmogorov, V., Boykov, Y., Rother, C.: Applications of parametric maxflow in vision. In: IEEE Int. Conf. on Computer Vision, Rio de Janeiro, Brasil (2007)
26. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts?. IEEE Trans. on Patt. Anal. and Mach. Intell. 24(5), 657–673 (2004)
27. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Comm. Pure Appl. Math. 42, 577–685 (1989)
28. Rother, C., Kolmogorov, V., Blake, A.: GrabCut: interactive foreground extraction using iterated graph cuts. ACM Trans. Graph 23(3), 309–314 (2004)
29. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. on Patt. Anal. and Mach. Intell. 22(8), 888–905 (2000)
30. Strang, G.: Maximal flow through a domain. Mathematical Programming 26(2), 123–143 (1983)
31. Vogiatzis, G., Torr, P., Cippola, R.: Multi-view stereo via volumetric graph-cuts. In: Int. Conf. on Computer Vision and Pattern Recognition, pp. 391–399 (2005)

Image Feature Extraction Using Gradient Local Auto-Correlations

Takumi Kobayashi and Nobuyuki Otsu

National Institute of Advanced Industrial Science and Technology,
1-1-1 Umezono, Tsukuba, Japan
{takumi.kobayashi,otsu.n}@aist.go.jp

Abstract. In this paper, we propose a method for extracting image features which utilizes 2 nd order statistics, i.e., spatial and orientational auto-correlations of local gradients. It enables us to extract richer information from images and to obtain more discriminative power than standard histogram based methods. The image gradients are sparsely described in terms of magnitude and orientation. In addition, normal vectors on the image surface are derived from the gradients and these could also be utilized instead of the gradients. From a geometrical viewpoint, the method extracts information about not only the gradients but also the curvatures of the image surface. Experimental results for pedestrian detection and image patch matching demonstrate the effectiveness of the proposed method compared with other methods, such as HOG and SIFT.

1 Introduction

Extracting features from an image is a fundamental procedure for various tasks, e.g., face or human detection [1,2], image patch matching [3], object recognition [4] and image retrieval [5]. It is important to extract characteristics of target objects and textures with retaining robustness to irrelevant variations resulting from environmental changes, such as changes in illumination or target position. Strictly speaking, we can identify two types of image features by focusing on image alignments: a shift-invariant type and a local image descriptor type.

The former type needs object regions not to be aligned and thus has the property of shift-invariance for the target objects. Fourier transformation and histogram based methods are traditionally applied to this type. This property of shift-invariance is particularly favorable for the task of object recognition, since it can then be carried out irrespective of the target position. However, it is difficult to obtain sufficient discriminative power for this type of features.

The latter type assumes aligned object regions and it is often dealt with in terms of a local image descriptor [3], which takes advantage of spatial alignment in the image region. The features of this type have been successfully developed and they play important roles, especially for image patch matching problems. These features include small patch [6], Shape Context [7], self similarity [8] and image gradients [9]. A comprehensive survey of local image descriptors is given in [3]. These local descriptors have been recently utilized in bag-of-feature frameworks which work particularly well for object recognition [10,4,11]. On the other hand, the shift-invariant features mentioned above

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 346–358, 2008.

can be naturally applied to local descriptors by simply dividing regions into several subregions (spatial binning), as in SIFT [12] and HOG [13].

In this paper, we propose a method for extracting shift-invariant image features which can also be applied as local descriptors. It extracts richer information, i.e., 2 nd order statistics of gradients, and thus obtains more discriminative power than standard histogram based methods. The proposed method is based on spatial and orientational auto-correlations of local image gradients: Gradient Local Auto-Correlation (GLAC). In GLAC, the image gradients are described sparsely in terms of their magnitude and orientation. Furthermore, the gradients can be extended to normal vectors on the image surface, which can be utilized for Normal Local Auto-Correlation (NLAC). We applied the proposed methods, GLAC and NLAC, as local image descriptors to two tasks: human detection and image patch matching. The experimental results demonstrate their effectiveness compared with other methods, including SIFT and HOG.

2 Related Work

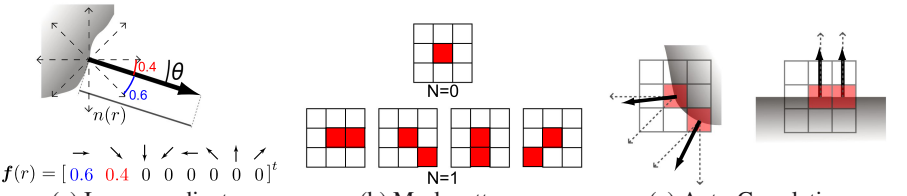
We mention here only closely related work. SIFT [12] and HOG [13] are some of the most successful features based on histograms of gradient orientations weighted by gradient magnitudes. These two methods slightly differ in the type of spatial bins that they use; HOG has a more sophisticated way of binning. The concept of correlation has also been adopted in self similarity [8], in which extracted edges in Shape Context [7] are substituted with cross-correlation values between local patches at a reference position and its local neighborhoods. Our work is most closely related to ECM [14] which utilizes joint histograms of orientations of gradient pairs. Differences in the details are described in Sec.3.2.

3 Gradient Local Auto-Correlations

In this section, we describe the details of the proposed method, Gradient Local Auto-Correlations (GLAC). It can be interpreted as a natural extension of HOG or SIFT from 1 st order statistics (i.e., histograms) to 2 nd order statistics (i.e., auto-correlations). In GLAC, image gradients are sparsely described in terms of their magnitudes and orientations. The proposed formulation extends naturally from Higher-order Local Auto-Correlation (HLAC) [15] of pixel values so as to deal with gradients as well. Therefore, GLAC inherits the desirable properties of HLAC for recognition: *shift-invariance* and additivity.

3.1 Definition of GLAC

Let I be an image region and $\boldsymbol{r}=(x,y)^t$ be a position vector in I. The image gradient $(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})^t$ at each pixel can be rewritten in terms of the magnitude $n=\sqrt{\frac{\partial I}{\partial x}^2+\frac{\partial I}{\partial y}^2}$ and the orientation angle $\theta=\arctan(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})$. As shown in Fig. 1(a), the orientation θ is coded into D orientation bins by voting weights to the nearest bins, and is described as a sparse vector $\boldsymbol{f}(\in \boldsymbol{R}^D)$, called the *gradient orientation vector* (in short, G-O vector).



(a) Image gradient (b) Mask patterns (c) Auto-Correlation

Fig. 1. Image gradients are described by the G-O vectors, together with the gradient magnitudes (a). Then, by applying mask patterns (b), auto-correlations of G-O vectors are calculated, weighted by the gradient magnitudes (c).

It is important that the image gradients are represented in terms of such quantized and sparse descriptors.

By using the G-O vector $\boldsymbol{f}$ and the gradient magnitude n, the N^{th}order auto-correlation function of gradients in local neighbors is defined as follows:

$$R(d_0, \cdots, d_N, \boldsymbol{a}_1, \cdots, \boldsymbol{a}_N) = \int_I w\big[n(\boldsymbol{r}), n(\boldsymbol{r}+\boldsymbol{a}_1), \cdots, n(\boldsymbol{r}+\boldsymbol{a}_N)\big] f_{d_0}(\boldsymbol{r}) f_{d_1}(\boldsymbol{r}+\boldsymbol{a}_1) \cdots f_{d_N}(\boldsymbol{r}+\boldsymbol{a}_N) d\boldsymbol{r}, \tag{1}$$

where $\boldsymbol{a}_i$ are displacement vectors from the reference point $\boldsymbol{r}$, f_d is the d-th element of $\boldsymbol{f}$ and w is a (scalar) weighting function, e.g., $\min$. Displacement vectors are limited to local neighbors because local gradients are supposed to be highly correlated.

Eq.(1) contains two kinds of correlations of gradients: *spatial* correlations derived from displacement vectors $\boldsymbol{a}_i$ and *orientational* correlations derived from the products of the element values f_{d_i}. We do not correlate image gradients themselves but G-O vectors which are quantized and represented sparsely. This is due to the empirical fact that, in HLAC [15], the auto-correlations of binary values, i.e., quantized data, are better for establishing recognition than those of the pixel values themselves. The function w composed of magnitudes n functions as the weights of the auto-correlation.

In practice, Eq.(1) can take so many forms by varying the parameters $N, \boldsymbol{a}_i$, and the weight w. In this paper, these are restricted to vary as follows: $N \in \{0, 1\}$, $a_{1x,y} \in \{\pm\Delta r, 0\}$, and $w(\cdot) \equiv \min(\cdot)$. The order of auto-correlation, N, is low, which enables extraction of sufficient geometric characteristics together with local displacements $\boldsymbol{a}_i$. The displacement intervals are the same in both horizontal and vertical directions due to isotropy of the image. We adopt $\min$ for w in order to possibly suppress the effect of isolated noise on surrounding auto-correlations. Thus, the practical formulation of GLAC is given by

$$\textbf{0}^{\textbf{th}}\textbf{order} \quad R_{N=0}(d_0) = \sum_{\boldsymbol{r}\in I} n(\boldsymbol{r}) f_{d_0}(\boldsymbol{r}) \tag{2}$$

$$\textbf{1}^{\textbf{st}}\textbf{order} \quad R_{N=1}(d_0, d_1, \boldsymbol{a}_1) = \sum_{\boldsymbol{r}\in I} \min\big[n(\boldsymbol{r}), n(\boldsymbol{r}+\boldsymbol{a}_1)\big] f_{d_0}(\boldsymbol{r}) f_{d_1}(\boldsymbol{r}+\boldsymbol{a}_1).$$

The configuration patterns of $(\boldsymbol{r}, \boldsymbol{r}+\boldsymbol{a}_1)$, i.e., the spatial auto-correlation patterns, are shown in Fig. 1(b). It should be noted that we obtain only four independent patterns

Algorithm 1. GLAC computation

Preprocessing: The G-O vector $\boldsymbol{f}$ and the gradient magnitude n are calculated from image gradients.
0 th order: At each pixel $\boldsymbol{r}$, summation in Eq.(2) is applied to only *two* non-zero elements of $\boldsymbol{f}$ with weight n.
1 st order: At each pixel $\boldsymbol{r}$, for each mask pattern (Fig. 1(b)), summation of products in Eq.(2) are applied to non-zero elements of $\boldsymbol{f}(\boldsymbol{r})$ and $\boldsymbol{f}(\boldsymbol{r}+\boldsymbol{a}_1)$ with weight of $\min[n(\boldsymbol{r}), n(\boldsymbol{r}+\boldsymbol{a}_1)]$. This takes only *four* times operations of multiplication.

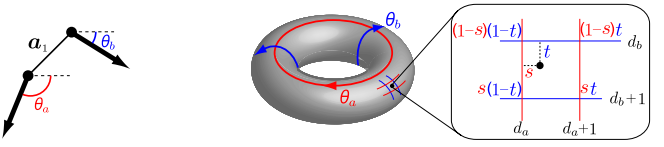
for 1 st order GLAC by eliminating duplicates which arise from shifts. For the 1 st order, the element values of G-O vector pairs determined by the mask patterns are multiplied and summed over the image (Fig. 1(c)). Although GLAC has high dimensionality ($D + 4D^2$), the computational cost is not large due to the sparseness of $\boldsymbol{f}$ (see Algorithm 1). Moreover, the computational cost of GLAC is invariant with respect to the number of orientation bins, D, since the sparseness of $\boldsymbol{f}$ is invariant with respect to D. In the case of calculating features in many sub-regions of an image, we can apply a method similar to the *integral image* approach [1], which is particularly effective for the object detection problem, e.g., face or pedestrian detection.

3.2 Interpretation

Histogram. While 0 th order GLAC simply corresponds to a histogram of gradient orientations used in SIFT [12] and HOG [13], 1 st order can be interpreted as a joint histogram of orientation pairs. Now, we consider the joint distribution of orientation pairs of local gradients, taking into account the fact that the orientation angles are periodic in $[0, 2\pi)$. Given a certain displacement vector $\boldsymbol{a}_1$ which determines the local pairs (Fig. 2(a)), the orientation pairs are jointly distributed on the torus manifold defined by the paired angles (Fig. 2(b)). The 1 st order GLAC corresponds to the joint histogram calculated by quantizing the distribution into $D \times D$ bins on the torus with *bilinear* weighting (Fig. 2(b)). This joint histogram weighted by w forms 2 nd order statistics naturally extended from the histogram of orientations (1 st order statistics). From this perspective, the 0 th order GLAC is a marginal histogram of the 1 st order. This suggests that the 0 th order components are not independent of the 1 st order and may be redundant, which is verified by experiments (see Sec.5).

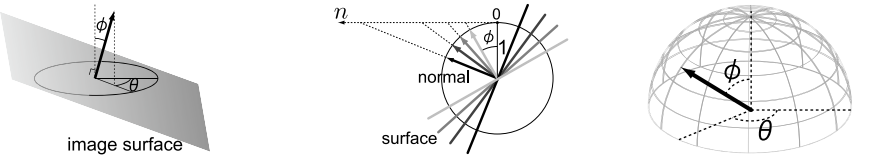
ECM [14] also utilizes a joint histogram of orientation pairs, but it is a special case of GLAC: $w \equiv 1$ (no weighting) and the G-O vector consists of binary values (0 or 1) in ECM. It suffers from boundary effects of the magnitude and the orientation of image gradients. Moreover, the displacement vectors are not specified in ECM, whereas they are determined according to the auto-correlation scheme in this paper.

Geometry. The 1 st order GLAC characterizes curvatures of image contours. The curvatures are quantized and patterned by the combinations of the orientations of local gradient pairs in the mask pattern as shown in Fig. 1(c). GLAC extracts image features in terms of gradients and curvatures which are fundamental properties of the image contours. In GLAC, the curvatures are distinguished by rotation. Rotational invariance,



(a) Gradient pair (b) Joint histogram on torus manifold

Fig. 2. GLAC is a joint histogram of paired angles on the torus manifold (b) determined by a displacement vector (a)



(a) Normal vector (b) Steepness of image surface (c) Hemisphere

Fig. 3. Normal vector to image surface

however, can be rendered by simply summing up the component values associated with curvature patterns which are matched by rotation.

Next, we consider the image surface defined by pixel values in 3-D space denoted as $\boldsymbol{z} = (x, y, I(x, y))^t$. The normal vector to the surface is calculated as follows:

$$\frac{\partial \boldsymbol{z}}{\partial x} \times \frac{\partial \boldsymbol{z}}{\partial y} = \left(-\frac{\partial I}{\partial x}, -\frac{\partial I}{\partial y}, 1\right)^t, \quad \phi = \arctan \sqrt{\left[\frac{\partial I(x,y)}{\partial x}\right]^2 + \left[\frac{\partial I(x,y)}{\partial y}\right]^2}, \qquad (3)$$

where ϕ is the angle of elevation (Fig. 3(a)). Thus, the gradient magnitude n determines the steepness of the local surface (Fig. 3(b)). The weight w controlled by n corresponds to the magnitude of the curvature on the image surface in 3-D space and, consequently, GLAC focuses on principal curvatures by means of weightings.

4 Normal Local Auto-Correlations

The normal vectors (Fig. 3) can be employed instead of the gradients described in Sec.3.1. The normals characterize the image surface in 3-D space while the gradients do the same for the image contours in a 2-D image plane. Thus, by using normals, Normal Local Auto-Correlation (NLAC) can be developed to extract the detailed features of the image surface, in a manner similar to GLAC.

4.1 Normal Orientation Vector

As shown in Fig. 3(a) and Eq.(3), a normal vector is characterized by the orientation θ in the x-y plane and the angle of elevation ϕ. The normal can be coded by *bilinear* weighting on the hemisphere composed of two angles θ, ϕ (Fig. 3(c)) and then the *normal orientation vector* (N-O vector) $\boldsymbol{g}$ can be defined in a manner similar to the G-O vector in Sec.3.1.

Here, the problem is how to define the scale of pixel values $I(x, y)$ in Eq.(3). The scale of $\partial I(x, y)$ (the pixel value domain), which is arbitrarily defined by users, e.g., $[0, 1]$ or $[0, 255]$, is intrinsically different from that of $\partial x, \partial y$ (the pixel location domain). Let a pixel value be I_o in certain scale, e.g., $[0, 1]$. The elevation angle ϕ in Eq.(3) can be rewritten as

$$\phi = \arctan\left(k\sqrt{\left[\frac{\partial I_o(x,y)}{\partial x}\right]^2 + \left[\frac{\partial I_o(x,y)}{\partial y}\right]^2} \right) = \arctan(kn) \tag{4}$$

where k is a scaling factor. The problem is how to determine k appropriately so as to be consistent with $\partial x, \partial y$.

The scaling k determines the distribution of normals on the hemisphere: if $k \to 0$, the normals would be concentrated near the zenith and, contrarily, if $k \to \infty$ they would be located only around the periphery. From the viewpoint that the normals are coded into equally spaced bins on the hemisphere and are described as the N-O vector, the scaling k can be determined so that the distribution of the normals is uniform along ϕ in order to make all bins on the hemisphere useful. In this case, the distribution is transformed by the function $\arctan(kn)$. In terms of histogram equalization [16], $\arctan(kn)$ is required to be similar to the probability distribution function of gradient magnitude n in order to make uniform distribution on ϕ. Thus, the scaling k is determined as

$$k = \arg\min_{k,l} \left| P(n) - l \arctan(kn) \right|^2 \tag{5}$$

where P is the probability distribution function of n and l is introduced so as to fit the ranges of P and $\arctan$, which does not affect the distribution.

4.2 Definition of NLAC

By using the N-O vector $\boldsymbol{g}$, NLAC can be computed as

$$R_{N=0}(d_0) = \sum_{\boldsymbol{r} \in I} g_{d_0}(\boldsymbol{r}), \quad R_{N=1}(d_0, d_1, \boldsymbol{a}_1) = \sum_{\boldsymbol{r} \in I} g_{d_0}(\boldsymbol{r}) g_{d_1}(\boldsymbol{r} + \boldsymbol{a}_1) d\boldsymbol{r}. \tag{6}$$

This does not include weighting whereas the weight derived from the gradient magnitude n is utilized in GLAC (Eq.(1)). This is because the N-O vector $\boldsymbol{g}$ already contains information about the magnitude in the angle of elevation ϕ. The computational cost of NLAC is small due to the sparseness of $\boldsymbol{g}$ as well as that of GLAC.

From a geometrical viewpoint, NLAC is a histogram of patterns of curvatures on the image surface. Although, in GLAC, the principal curvatures on the surface are highly weighted, all patterns of the curvatures can be captured in NLAC regardless of the magnitudes of the curvatures. For example, a curvature which includes a zero gradient, e.g., Fig. 4(a), is disregarded in GLAC, but it is counted in NLAC. However, the count of a flat curvature, e.g., Fig. 4(b), is closely related to the area size of the object. This is a square function of the target scale, which reduces robustness to the scale. Therefore, when the target scale is not normalized, we disregard curvature patterns arising from flatness which are related to only one element of $\boldsymbol{g}$ associated with the zenith bin on the hemisphere. The number of disregarded patterns is 1 (0^{th}order) + 4 (1^{st}order) = 5.

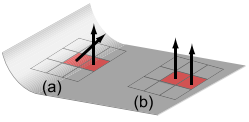
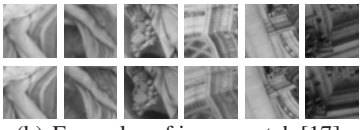


Fig. 4. NLAC can capture various patterns of curvatures, even those containing zero gradients: (a) foot of hill and (b) flatness. The curvature pattern of (b) would be disregarded.

(a) Examples of person image [13]



(b) Examples of image patch [17]

Fig. 5. Images in datasets

5 Experimental Results

We apply the proposed methods to two kinds of task: human detection [13] and image patch matching [17] in order to compare the performances with those of HOG [13] and SIFT [12] which have been some of the most successful methods for these tasks.

5.1 Human Detection

In this experiment, the extracted features are classified by using the linear SVM [18]. The proposed methods were tested on the INRIA person dataset (Fig. 5(a)), details of which are in [13]. We selected 2416 person and 12180 person-free images (64×128) for training, and 1132 person and 13590 person-free images for testing. For quantifying and comparing detectors, we plotted Receiver Operating Characteristics (ROC) curves by calculating False Positive (FP) and True Positive (TP) Rates.

Although GLAC and NLAC are completely shift-invariant, the detection problem does not require this property due to roughly aligned person images arising from shifting the detection window in the image. Thus, for accuracy comparisons, the image region is divided into sub-regions (blocks), e.g., 4×4 blocks, and the GLAC/NLAC features extracted in these blocks are integrated into a final feature vector in the same manner as SIFT [12]. Spatial binning reduces shift-invariance but increases discriminative power as shown in the next.

Comparison to the other methods. First, we compare overall performances of the proposed methods with those of the other methods: HOG [13], Steerable Filter [19,17] and Steerable Filter Local Auto-Correlation (SLAC). HOG, for which the parameter settings are those of [13], has produced the best performance for this database. The Steerable Filter feature consists of the rectified response values of fourth order derivative filters [19], and this method has worked well in image patch matching [17]. SLAC is newly constructed here by using the Steerable Filter feature vector instead of $\boldsymbol{g}$ in Eq.(6). In the Steerable Filter and SLAC approaches, spatial binning is also applied. The

performance results are shown in Fig. 8(a). Both the GLAC and NLAC methods outperform the other methods including HOG. The performance of NLAC is lower than that of GLAC even when utilizing the same spatial bins, and these methods are compared in the last part of this section. GLAC with 4×5 blocks has a higher dimensionality than GLAC with 3×4 blocks, but results in further improvement. When 3×4 spatial bins are utilized, the dimension of GLAC features is almost the same as that of HOG. Note that the number of these spatial bins is significantly smaller than for HOG and thus larger spatial perturbations can be allowed. The performance of SLAC is a great improvement on that of Steerable Filter, but it is inferior to GLAC. This is because the G-O vector is much sparser than the Steerable Filter vector. As described before, auto-correlations work particularly well for sparse data.

Performance Study. Next, focusing on GLAC, we give details of the parameter settings and their effects on performance. We refer to the baseline parameter settings as: 1) the Roberts gradient filter; 2) 9 orientation bins in 360 degrees; 3) a spatial interval $\Delta r = 1$; 4) a weighting $w(\cdot) \equiv \min(\cdot)$; 5) only 1 $^{\text{st}}$order auto-correlation; 6) block-wise L2-Hys normalization; 7) 3×4 spatial blocks, which are the same as in Fig. 8(a).

[Gradient] Gradient computation is the first processing step that may affect the final performance. We applied three types of filters: Roberts, Sobel and one-dimensional derivatives ($[-1, 0, 1]$). The Roberts filter, which is the most compact, is most effective, whereas the smoothed Sobel filter is least effective (Fig. 8(b)). As shown in [13], smoothing the images results in reduced performance.

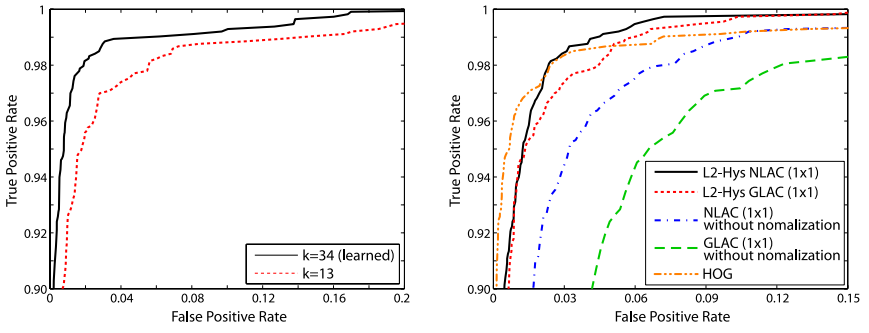
[Orientation bins] Orientation bins are evenly spaced over $[0^\circ, 180^\circ)$ (unsigned gradients) or $[0^\circ, 360^\circ)$ (signed gradients). Fig. 8(c) shows that finer binning increases performance. Contrary to the results in [13], the signed gradient works even better than the unsigned gradient. For auto-correlations of orientations, signed gradients seem to be preferable. The recent results of object recognition using HOG have also shown a similar tendency [11].

[Spatial interval] The only parameter in auto-correlations is the spatial interval Δr which is closely related to the scale of the objects to be recognized. As shown in Fig. 8(d), small interval values, really *local* auto-correlations, work well with the compactness of the Roberts filter.

[Weighting] The weight w in Eq.(1) is qualitatively defined as $\min$, taking the perspective of noise reduction. It is quantitatively compared with $\max$ and product ($\prod n$) in Fig. 8(e). As expected, $\min$ is the best, due to the noise reduction effect.

[Correlation order] The composition of GLAC can be varied as follows: only 1 $^{\text{st}}$order, both 0 $^{\text{th}}$and 1 $^{\text{st}}$order, and only 0 $^{\text{th}}$order. Fig. 8(f) shows that the addition of the 0 $^{\text{th}}$order to the 1 $^{\text{st}}$order has no, even worse, effect on performance. Thus, the 0 $^{\text{th}}$order components seem to be redundant, as suggested in Sec.3.2.

[Normalization] We adopted two types of normalization: L2 and L2-Hys. L2 refers to normalization by L2-norm and L2-Hys means clipping component values after L2 as in [12]. These normalizations are applied either to whole feature vector or block-wise. Fig. 8(g) shows that L2-Hys outperforms L2 while block-wise normalization is better



(a) Scaling factor (b) Comparison without spatial binning

Fig. 6. The detection performances of NLAC with various settings. Details are in the text.

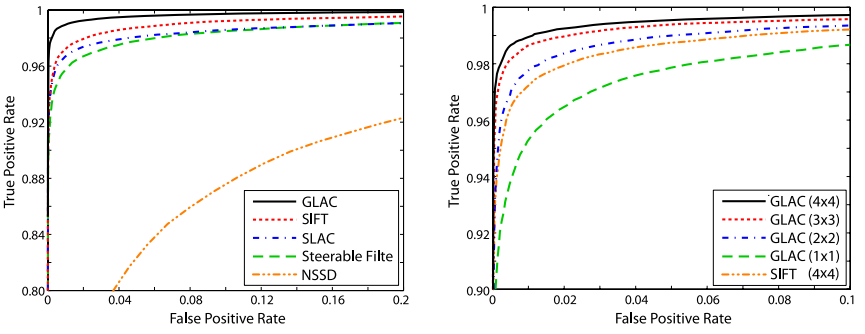
than whole normalization. In summary, block-wise L2-Hys normalization is the best, and performance is greatly improved, compared to performance without normalization.

[Spatial bins] Due to the dimensionality of GLAC, we applied somewhat coarser spatial binning, equally spaced over an image (64×128) as in [12]. As shown in Fig. 8(h), binning finer than 3×4 results in sufficiently good performance, and, in particular, 4×5 binning is most effective. Spatial binning results in greatly improved performance, compared to performance without spatial binning (1×1).

NLAC. In NLAC, the scaling factor k in Eq.(4) is learned from the MIT pedestrian dataset [20]; k=34. Fig. 6(a) shows that the learned value of k is appropriate and effective compared with a randomly chosen value of k=13. In Fig. 8(a), NLAC of 3×4 blocks outperforms HOG but it is inferior to GLAC. On the contrary, for no spatial binning (1×1) in Fig. 6(b), the performance of NLAC with L2-Hys is superior to that of GLAC. The effect of normalization (L2-Hys) on performance is greater for GLAC than for NLAC, by comparison of the results without normalization. In NLAC, the gradient magnitude n is already transformed by $\arctan$ in Eq.(4) at each pixel, which reduces the effect of L2-Hys normalization as a nonlinear operation. It is noteworthy that, even when other processes (spatial binning and normalization) are not applied, NLAC gives high performance with retaining the favorable properties of shift-invariance and additivity. In summary, GLAC is better suited to local descriptors and NLAC is better suited to shift-invariant features.

5.2 Image Patch Matching

We applied the proposed method (GLAC) to local image descriptors for image patch matching on the database of image patches [17] (Fig. 5(b)). This database contains matched image patches (64×64) collected by using SIFT detector and descriptor [12] and further 3D point estimation from tourist photographs of Trevi Fountain, Yosemite Valley and Notre Dame. See [17] for details of the database. We followed the procedure in [17] for training and testing: 10,000 matched pairs and 10,000 non-matched



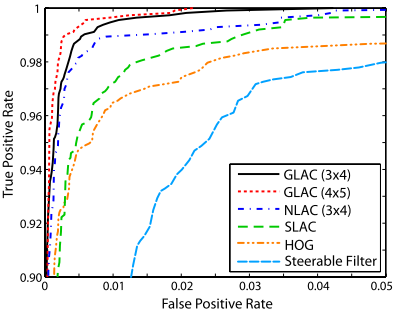
(a) Overall comparison (b) Spatial binning

Fig. 7. The results of image patch matching. Details are in the text.

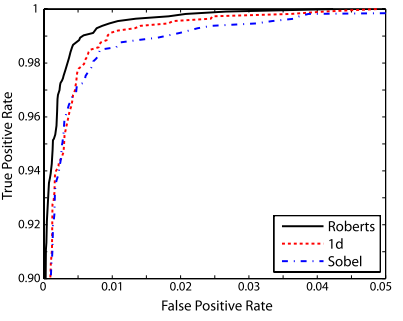
pairs were randomly sampled from the Trevi and Yosemite datasets in order to learn parameters for the local image descriptors. For testing, 50,000 matched and 50,000 non-matched pairs were also randomly chosen from the Notre Dame dataset.

The image descriptors were feature vectors extracted from image patches as in the method for human detection above. Image patch pairs for which the descriptor vectors were sufficiently close were classified as matched. We computed the Euclidean distance between descriptors of image patch pairs and then, for evaluating performance, an ROC curve was constructed, based on the two histograms of the distances for all true matching and non-matching cases in the dataset. In the learning phase, the parameters of the descriptors were appropriately determined according to the evaluation results of the ROC for the training dataset; minimizing the FP rate when the TP rate is 0.98. After descriptors were learned, performances were evaluated on the test dataset.

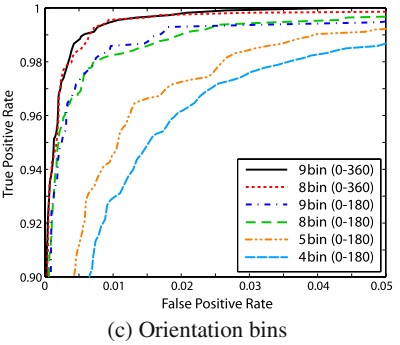
The image descriptor was constructed as follows: First, the image patch was smoothed by the Gaussian kernel of the standard deviation σ, and then the feature was extracted with spatial binning. Finally, L2-Hys of the threshold γ was applied to whole feature vector. In the proposed method, the 0^{th}order and 1^{st}order components were weighted by μ and $(1-\mu)$, respectively, for calculating distances between descriptors. We applied only GLAC of the orientation bin $D=8$ according to the comparison between GLAC and NLAC in Sec.5.1, and GLAC is compared with the other methods: SIFT [12], Steerable Filter [17], SLAC, and the normalized sum of squared differences (NSSD). The parameters to be learned were $\sigma, \Delta r, \gamma$ in GLAC and SLAC while those of Steerable Filter were σ, γ. The parameters of SIFT and NSSD were set to the values described in [17]. Unlike [17], spatial binning of all methods was not learned but constant (4×4) in order to accurately compare the performance of the feature extraction methods themselves. Fig. 7(a) shows the results. GLAC outperformed the other methods including SIFT. It is noteworthy that, in all experiments, the learned weight μ of the 0^{th}order was 0 and so the 0^{th}order components are redundant for image patch matching as well as for human detection. The learned value of σ was non-zero and it is found that preprocessing of smoothing images contributes to an improvement in contrast to human detection. Furthermore, the spatial interval learned was larger ($\Delta r \sim 10$), thus implying a stronger effect for somewhat broader texture alignment. The performance of GLAC



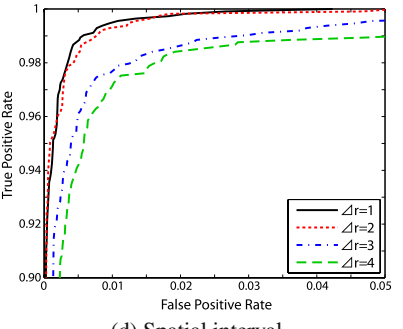
(a) Overall performance



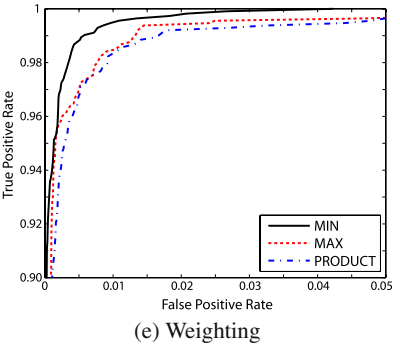
(b) Gradient method



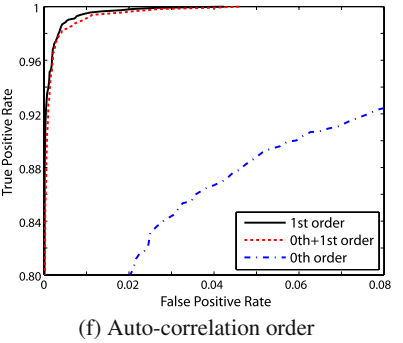
(c) Orientation bins



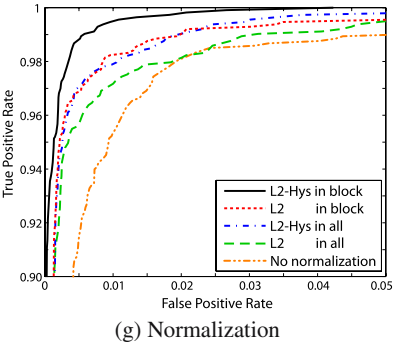
(d) Spatial interval



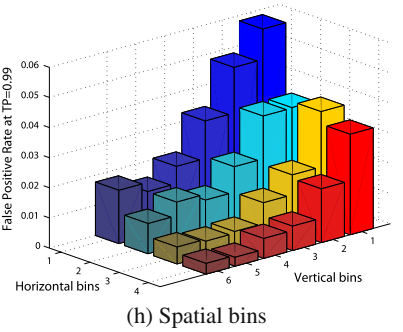
(e) Weighting



(f) Auto-correlation order



(g) Normalization



(h) Spatial bins

Fig. 8. The detection performances of GLAC with various settings. Details are in the text.

along with spatial binning is shown in Fig. 7(b). Finer binning than 2×2 produced a superior result to SIFT with 4×4 binning.

6 Conclusions

We have proposed two methods for extracting image features: Gradient Local Auto-Correlation (GLAC) and Normal Local Auto-Correlation (NLAC). This framework is based on spatial and orientational auto-correlations of local image gradients and normals, which renders shift-invariance and additivity as in HLAC [15]. The gradient is sparsely described in terms of magnitude and orientation for GLAC. The gradients can be extended to normal vectors on the image surface for NLAC. These methods extract local geometrical characteristics of the image surface in more detail than standard histogram based methods, since 2 ndorder statistics are utilized. In experiments for human detection and image patch matching, the proposed methods produced favorable results compared with the other methods. It was also found that GLAC works well with spatial binning and normalization, although shift-invariance is lost, whereas NLAC without these processings is suitable for shift-invariant recognition problems.

References

1. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 511–518 (2001)
2. Leibe, B., Seemann, E., Schiele, B.: Pedestrian detection in crowded scenes. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 878–885 (2005)
3. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. Pattern Analysis and Machine Intelligence 27, 1615–1630 (2005)
4. Lin, Y.Y., Liu, T.L., Fuh, C.S.: Local ensemble kernel learning for object category recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
5. Smeulders, A.W., Worring, M., Sntini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. Pattern Analysis and Machine Intelligence 22, 1349–1380 (2000)
6. Boiman, O., Irani, M.: Detecting irregularities in images and in video. In: International Conference on Computer Vision, pp. 462–469 (2005)
7. Belongie, S., Malik, J., Puzicha, J.: Matching shapes. In: International Conference on Computer Vision, pp. 454–461 (2001)
8. Shechtman, E., Irani, M.: Matching local self-similarities across images and videos. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 511–518 (2007)
9. Laptev, I., Lindeberg, T.: Space-time interest points. In: International Conference on Computer Vision, pp. 432–439 (2003)
10. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. International journal of computer vision 73, 213–238 (2007)
11. Bosch, A., Zisserman, A., Munoz, X.: Image classification using random forests and ferns. In: International Conference on Computer Vision, pp. 1–8 (2007)
12. Lowe, D.: Distinctive image features from scale invariant features. International Journal of Compuater Vision 60, 91–110 (2004)

13. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 20–25 (2005)
14. Rautkorpi, R., Iivarinen, J.: A novel shape feature for image classification and retrieval. In: International Conference on Image Analysis and Recognition, pp. 753–760 (2004)
15. Otsu, N., Kurita, T.: A new scheme for practical flexible and intelligent vision systems. In: IAPR Workshop on Computer Vision (1988)
16. Russ, J. (ed.): The Image Processing Handbook. CRC Press, Boca Raton (1995)
17. Winder, S., Brown, M.: Learning local image descriptors. In: Computer Vision and Pattern Recognition, pp. 1–8 (2007)
18. Vapnik, V. (ed.): Statistical Learning Theory. Wiley, Chichester (1998)
19. Freeman, W., Adelson, E.: The design and use of steerable filters. Pattern Analysis and Machine Intelligence 13, 891–906 (1991)
20. Mohan, A., Papageorgiou, C., Poggio, T.: Example-based object detection in images by components. Pattern Analysis and Machine Intelligence 23, 349–361 (2001)

Analysis of Building Textures for Reconstructing Partially Occluded Facades

Thommen Korah[1] and Christopher Rasmussen[2]

[1] HRL Laboratories, LLC Malibu, CA, USA
tkorah@hrl.com
[2] University of Delaware Newark, DE, USA
cer@cis.udel.edu

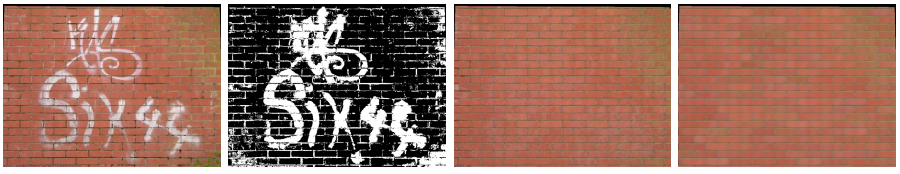
Abstract. As part of an architectural modeling project, this paper investigates the problem of understanding and manipulating images of buildings. Our primary motivation is to automatically detect and seamlessly remove unwanted foreground elements from urban scenes. Without explicit handling, these objects will appear pasted as artifacts on the model. Recovering the building facade in a video sequence is relatively simple because parallax induces foreground/background depth layers, but here we consider static images only. We develop a series of methods that enable foreground removal from images of buildings or brick walls. The key insight is to use *a priori* knowledge about grid patterns on building facades that can be modeled as Near Regular Textures (NRT). We describe a Markov Random Field (MRF) model for such textures and introduce a Markov Chain Monte Carlo (MCMC) optimization procedure for discovering them. This simple spatial rule is then used as a starting point for inference of missing windows, facade segmentation, outlier identification, and foreground removal.

1 Introduction

An important step in vision-based architectural modeling [1,2,3] is the creation of texture maps representing each planar section of a building's facade. A frequent complicating factor is the presence of other, unknown objects in the scene between the camera and building plane—e.g., trees, people, signs, poles, and other clutter of urban environments. In a similar class are objects reflected in building windows. Without explicitly recognizing and removing them, these foreground objects will be erroneously included in the building appearance model, as can be seen in the results of [4,5] among others. Many artifacts also arise due to the lack of strict constraints on parallelism, continuity of linear edges, and symmetry. With manual intervention one can cut out such features and replace them with nearby symmetric or repeated building features [6]. The larger problem that motivates this paper is whether and how foreground objects can *automatically* be eliminated.

Considering Fig. 1, two major obstacles to overcome for background recovery are (i) identifying the problem areas and, (ii) actually removing foreground objects to reveal the building structure behind them. Given a sequence captured by

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 359–372, 2008.



(a) (b) (c) (d)

Fig. 1. Virtual graffiti removal. (a) Original image; (b) Automatically detected foreground pixels (c) Tile-aligned exemplar-based inpainting (d) Eigenimage reconstruction. Tiles with $> 25\%$ outliers were sampled. While there is some loss of detail in (d), many local characteristics are retained.

a translating camera, parallax is an obvious cue to identify foreground objects at different depths [7]. However, even from a single image, humans are adept at "mentally scrubbing" away distracting elements and envisioning the appearance of the obliterated regions.

We make the simplifying assumption that the background is strongly structured and exhibits characteristics of a near-regular texture that dominates the image. This is frequently the case for close-up images of sections of buildings with brick patterns or window grids. The basic idea is that by discovering these textures automatically and collecting statistics on tile appearance, we can automatically segment foreground objects as texture outliers and either reconstruct them from or replace them with unoccluded patterns elsewhere in the texture. We demonstrate how these spatial priors enable subsequent operations such as inference of missing windows, facade segmentation, outlier identification, and foreground removal.

1.1 Previous Work

Modeling of urban and architectural environments has been studied for several years. Government agencies have traditionally used it for development planning or military strategy. The success of recent products like Google Earth, SketchUp, and Microsoft Virtual Earth has enabled urban modeling to be done in a distributed, voluntary, and "wikified" manner. However, the photogrammetric techniques used to compute the geometry of a scene can be brittle when dealing with large-scale and unconstrained urban scenes.

In this work, we ignore geometry and focus on recovering a "clean" mosaic of the facade that can subsequently be used as a texture map. With the exception of the MIT City Scanning Project [2] and the 3D City Model Generation work at Berkeley [5], none of the systems listed previously address issues of missing data, occlusions, perspective or other factors that degrade the visual quality of the texture maps. Even these two systems employ very crude methods of photometric blending and interpolation with little consideration of topology. There are no safeguards against wall pixels being copied over windows or other misalignment issues. Debevec's view-dependent texture mapping [1] gets around this issue

without explicitly handling the occluding elements. Once the problem areas are identified, inpainting techniques based on PDEs [8] or non-parametric exemplar methods [9,10], combined with some prior knowledge of the architectural domain, offer a principled way to remove large foreground elements.

We specifically try to interpret the building facade from a single static image. Dick at al. [3] were among the first to attempt automatic labeling of architectural elements. Mayer and Reznik [11] focused on facade interpretation using implicit shape models to extract windows. Although both methods used Markov Chain Monte Carlo (MCMC) [12] to simulate the posterior, there was no information about the connectivity between the detected elements. This makes high-level analysis and manipulation difficult. In graphics, split grammars were introduced by Wonka [13] to formally describe the derivation of architectural shapes for procedural modeling. A few researchers combined the grammar-based procedural modeling with the concept of *parsing* images of buildings [14], although inconsistencies and occlusion cause these systems to fail. Recently, Mueller [15] presented an impressive interactive system that takes a single rectified image of a building as input and computes a 3D geometric and semantic model with much greater visual quality and resolution.

The above methods use very specialized models and show examples on a restricted set of images. Like us, they require that the facades contain repetitive elements (typically windows) exhibiting regularity. However, the input needs to be clipped and rectified before any processing. The case of detecting occluding elements or seeing through them is seldom handled, primarily because of stringent assumptions on the nature of symmetric patterns. Instead of tuned window detectors, we develop a generic grouping framework to detect near-regular lattice structures. Texture-specific models (such as those for windows under perspective) can be easily incorporated, while still being robust to occlusions and small irregularities. Reliably discovering the underlying symmetry in texture and structure is crucial for facade analysis and reconstruction.

Approaches based on RANSAC [16] and the cascaded Hough transform [17] have been used to find regular, planar patterns. However, many buildings in our test set do not exhibit the "checkerboard" style consistency these methods require. Our definition of a lattice structure is derived from the literature on Near-Regular Textures (NRT) [18,19]. An NRT is a geometric and photometric deformation from its regular origin of a congruent wallpaper pattern formed by 2D translations of a single tile or *texel*. Any warped 4-connected lattice constitutes an NRT. An iterative algorithm for NRT discovery by higher-order correspondence of visually similar interest points was described by Hays et al. in [20]. Besides being computationally expensive, directly applying their method might retrieve tiles which do not correspond exactly to semantically meaningful units. A technique to extract texels from homogeneous, 2.1D, planar texture with partial occlusion was presented in [21], but assumed that the placement of the texels was statistically uniform without any global structure. Bayesian approaches based on Markov Random Fields [22] have also been utilized for localizing grid structures in genome sequencing [23]. We adopt a similar approach

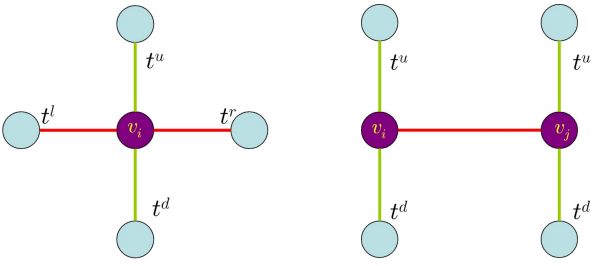


Fig. 2. MRF (a) local node and (b) clique potential "neighbor" vectors

that more generally applies to many different types of NRTs. Lin and Liu [24] used an MRF model to track dynamic deformable lattices, but assumed that the texels had already been discovered,

The next section describes our MRF/MCMC approach for efficient NRT discovery. We then describe a series of methods, woven together by the common goal of background recovery, to extract additional properties of the facade. This kind of information efficiently computed on-board an autonomous platform could also assist in view planning and focus of attention control. Finally, results are shown on a variety of building images.

2 Discovering Building Texture

For brick images (Fig. 1), we used an efficient power spectrum approach [25] that although simple, worked well on a variety of images. This section describes our more general MCMC algorithm for discovering rectangular NRTs. Previous algorithms for texture discovery [16,17] have used interest point or corner detectors to demarcate potential texels. Since we are interested in rectangular shaped windows, straight lines are first detected in the input image. Similar to [26], rectangles are hypothesized from pairs of approximately parallel line segments resulting in hundreds of rectangles. Under perspective, each rectangle v_i is defined as $(p_k^0, p_k^1, p_k^2, p_k^3)$ denoting the 4 corners of a quadrilateral in anti-clockwise order with p_{k1} as the upper-left position. We do not represent them using vanishing points as in [26] to avoid estimation errors in earlier stages from cascading through the pipeline.

The process results in a couple of thousand rectangles for a typical image. Some amount of pruning can be done by sorting the rectangles based on mean gradient strength along its boundaries and removing ones that are not well aligned with the edges. We conservatively keep the top 700 such rectangles (the average number of windows in our images is 15). By overestimating this number, the grouping algorithm is allowed to recover the best possible lattice without using hard thresholds early in the pipeline. Other image discretization methods such as interest points, correlation peaks [20], or color segmentation may also be used to generate v_i.

Given the set of tokens $v_i \in V$, we construct a pairwise MRF $G = (V, E)$. Each token is a random variable that constitutes a node of the undirected graph G, with edges $e_{ij} \in E$ representing the dependency between v_i and v_j. Since the probability of the states of a texton in an NRT is only locally dependent, the MRF model naturally preserves this Markov property. While [24] exploited it for tracking by preserving the structure over time, our goal is to build up the initial grid by linking together image tokens that exhibit the lattice topology.

The solution involves gradually evolving the lattice configuration by iteratively adding and removing edges within an MCMC framework. At the end of the process, links are created along vectors $\mathbf{t}_i^o : o \in \{r, l, u, d\}$ to the most likely right, left, up and down neighbors of v_i (Fig. 2) without violating grid constraints. Similarly, $v_i^o : o \in \{r, l, u, d, NULL\}$ denote its neighbors, if any, in each direction. Since we do not assume that texels are tightly packed and adjacent in the image, each node can potentially be linked to several others, increasing the combinatorics of the problem. Given image I, we wish to obtain the MAP estimate for the graph configuration

$$p(G|I, T, S) \propto p(I|T, S, G)\ p(S|G)p(T|G)\ p(G). \tag{1}$$

The image likelihood $p(I|T, S, G)$ is encapsulated in the rectangle detection and is neglected here. Color histograms, proximity of rectangle boundaries to image edges, or learned appearance models are all possible likelihood models. The shape prior $p(S|G)$ can be used to favor known shape models, though here we set it to unity since we are only dealing with rectangles. The graph prior $p(G)$ models any global intuition about the nature of the grid or the degree of connectedness. We set this to be unity.

The topology prior $P(T|G)$ is represented as a pairwise MRF whose joint can be factored into a product of local node potentials Φ and clique potentials Ψ:

$$P(T|G) \propto \prod_i \Phi(v_i) \prod_{i,j \in E} \Psi(v_i, v_j).$$

To model a grid, we measure the symmetry of direction vectors from a node to its neighbors. Let $\delta(\mathbf{t}_1, \mathbf{t}_2) = e^{-\beta||\mathbf{t}_1 - \mathbf{t}_2||}$ be a similarity measure between two neighbor vectors assuming both edges are in G. The potentials are now defined as:

$$\Phi(v_i) = e^{-\gamma(4-n_i)} * \delta(\mathbf{t}_i^r, -\mathbf{t}_i^l) * \delta(\mathbf{t}_i^u, -\mathbf{t}_i^d), \tag{2}$$

$$\Psi(v_i, v_j) = \delta(\mathbf{t}_i^u, \mathbf{t}_j^u) * \delta(\mathbf{t}_i^d, \mathbf{t}_j^d) * \mathcal{B}(v_i, v_j). \tag{3}$$

where n_i denotes the degree of node v_i. Thus we encourage increased connectivity as well as left/right and up/down edge pairs to be 180 degrees apart with similar magnitudes. The interaction potential between horizontal neighbors forces their vertical edges to be approximately parallel. For missing edges, a small fixed value of 0.2 is assigned to δ. These functions effectively model the generic lattice configuration as will be shown.

The function $\mathcal{B}(v_i, v_j)$ is used to specify any texture-specific pair-wise relationships between the texels. For building images and windows under perspective, we

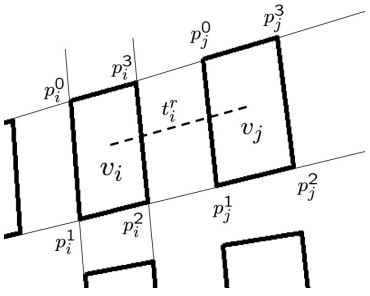


Fig. 3. Illustration of windows under perspective

incorporate constraints such as overlap, cross ratio, and appearance similarity. Using Fig. 3 to illustrate, we list various heuristics that reflect the probability of v_i and v_j being connected by a horizontal edge. The case of vertical neighbors is analogous.

- Windows on the same level have their bottom and top edges aligned with each other, implying that points $(p_i^0, p_i^3, p_j^0, p_j^3)$ and $(p_i^1, p_i^2, p_j^1, p_j^2)$ are collinear.
- One projective invariant is the cross ratio $Cross$. Assuming parallel window sides and negligible noise, the 4 upper and lower points in Fig. 3 should have approximately the same cross ratio. We define $CR = \left|1.0 - \frac{Cross(p_i^0, p_i^3, p_j^0, p_j^3)}{Cross(p_i^1, p_i^2, p_j^1, p_j^2)}\right|$ to quantify this measure. For horizontal neighbors, this essentially measures how parallel the vertical edges of the windows are.
- The horizontal dimensions of windows under perspective should vary smoothly on both the upper and lower sides.
- Windows and texels in general should not overlap.
- The corners of each polygon are correlated with each other to ensure appearance similarity. Correlating the entire window would be sensitive to occlusions and pose variations, while the corners are typically more distinctive. Formally, $XC(v_i, v_j) = \sum_{k=0}^{3} \frac{NCorr(Patch(p_i^k), Patch(p_j^k))}{4}$ where XC is the mean normalized cross correlation $NCorr$ of 11×11 patches centered at each of the 4 window corners.

These heuristics are converted into Gaussian likelihood functions that penalize deviations from our assumptions, and incorporated into $\mathcal{B}$. They are then used in conjunction with the generic lattice potentials defined in (2) and (3).

2.1 Optimization

We use a Markov Chain Monte Carlo (MCMC) framework to iteratively maximize (1), probabilistically adding and removing edges from the initial graph G_0 in a fashion similar to the multi-target tracking method of [27]. A Markov chain

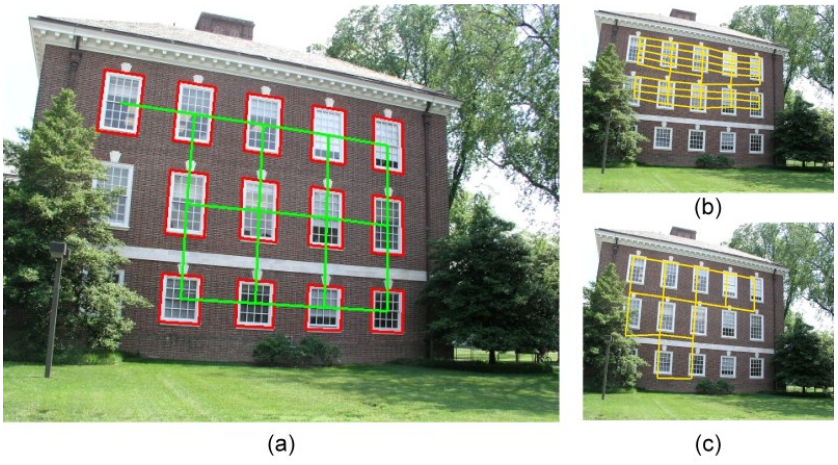


Fig. 4. (a) Result of our window grid discovery method; (b) reported best lattice from [20] which does not repeatably find the correct scale or centering while being less efficient; (c) best handpicked lattice from the various iterations of [20]

is defined over the space of configurations $\{G\}$ and the posterior is sampled using the Metropolis-Hastings [28] algorithm. A new state G'_t is accepted from state G_t with probability

$$p = \min(1, \tfrac{p(G'_t|I,T,S)q(G_t|G'_t)}{p(G_t|I,T,S)q(G'_t|G_t)}).$$

The graph G_0 is initialized by connecting each node with its lowest cost neighbor. The cost $E_{score}(i,j)$ for each pair of nodes v_i, v_j is measured as the total number of other nodes within a threshold distance of the line parametrized by the two nodes, scaled by rough shape similarity $\mathcal{B}$. Distracting elements or other deformations will cause inconsistencies in this Maximum Likelihood estimate; nevertheless, it provides a useful starting point for the MCMC simulation. Proposal updates $\mathcal{Q}(G'_t|G_t)$ for MCMC consist of edge additions or removals applied to a node v_k. Modifying edges one component at a time leads to better success rate for transitions. The transitions are made only in the up and right directions in order to keep the reverse transition probability simple. Two functions, picked probabilistically, govern how v_k is selected in each MCMC iteration: (i) an unguided scheme $\mathcal{Q}_1$ in which v_k is chosen uniformly from all nodes, and (ii) a guided hypothesis generation $\mathcal{Q}_2$ in which the edge is selected from a dynamic pool $\mathcal{P}_q$ of potentially good connections. As the grid converges to the correct solution, $\mathcal{Q}_2$ facilitates lattice growth and completion by hypothesizing edges close to the good parts of the lattice.

Let $e^o_{kl} : l \in \{1, \ldots, n_k\}$ be potential edges from v_k to its neighbors in direction o. In $\mathcal{Q}_1$, o is uniformly chosen from the up and right directions. The edge $\mathbf{t}^o_k$ from node v_k is turned off with fixed probability p_{off} or assigned a neighbor by sampling from $E_{score}(k,\cdot)$. Neighbors that seem to conform to the topological and visual priors are picked more often. Random selection alone can be inefficient in

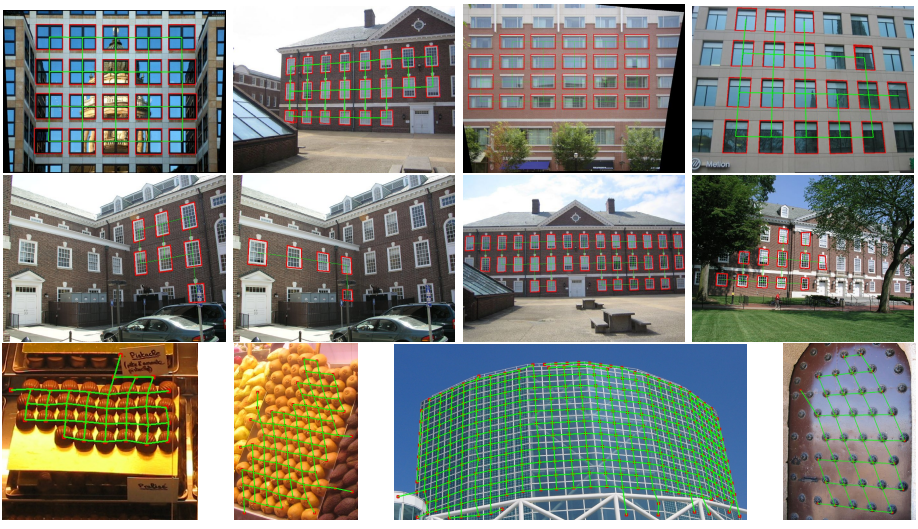


Fig. 5. Inferred lattice for various building images (top two rows). Bottom row shows results on a few images from the NRT database [29] by grouping purely on structure. Using a texture-specific $\mathcal{B}$ function could have prevented spurious links between dissimilar texels.

steering the optimization towards completing the lattice. When a node v_k is visited in $\mathcal{Q}_1$, its unbalanced edges, if any, are identified. A new link is hypothesized and added to priority queue $\mathcal{P}_q$ with a ranking function that encourages symmetry between opposite edges. A new proposal in $\mathcal{Q}_2$ simply involves removing the highest priority edge from $\mathcal{P}_q$ and adding it as an edge in state G'_t. The chain is *irreducible* because a series of edge additions and deletions can take the graph from one state to any other state. The stochastic elements also guarantee *aperiodicity* by not getting trapped in cycles. Together, they satisfy the MCMC conditions of ergodicity to ensure that the chain will converge to the stationary distribution.

2.2 Experimental Results

We show grouping results on building images as well as textures from the PSU NRT database [29]. The number of MCMC iterations was set to 10000 and the best MAP estimate chosen as the final configuration. A breadth-first traversal separates out the connected components. A user can then iterate over the larger ones to pick a best lattice, or the selection can be done automatically. For building images, we use edge alignment to rank each lattice. Figures 4(a) and 5 demonstrate our results on several images. The building images (top two rows) are characterized by occlusions, shadows, reflections, and variation among windows; purely appearance-based systems can be sensitive to these effects. Both windows as well as its topological structure in the form of a neighborhood graph has been captured. By only enforcing local smoothness in appearance and geometry, the grouping is robust to small changes in window dimensions and perspective effects.

Figure 4 compares the result of our method with the algorithm of Hays [20]. One of the main disadvantages of [20] is efficiency. Generating the results for an image took approximately 30 minutes. In contrast, our method takes less than a minute in total on a 1.6GHz Pentium M laptop. Execution times for our method on the image of Fig. 4a are (i) rectangle hypotheses - 20.2 sec (in Matlab), (ii) initial graph construction - 15.9 sec (C++), and (iii) MCMC grouping - 1.9 sec (C++). Rectangle hypotheses can be speeded up by porting to C++ while the main bottleneck in the graph construction is due to exhaustive searching among nodes for nearest neighbors. The correlation peaks used by [20] also suffer from the lack of centering on windows or other semantically meaningful aspects of the image. Finally, after repeated iterations, the best lattice that it eventually outputs based on their metric is not perceptually the best and requires handpicking.

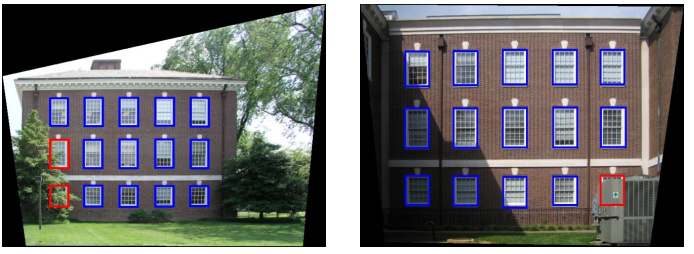
In order to test the adherence of our potential functions to the 4-connected definition of an NRT, we applied the grouping on pure NRTs after setting $\mathcal{B}$ to unity. Firstly, candidate tokens of like elements need to be extracted from the image. Similar to Hays, for images where MSER [30] gave a reasonably good initialization, these point features were grouped. If not, we applied the Matlab functions used by [20] to detect correlation peaks from a user-selected patch identifying a texel. This interaction only involves drawing a bounding box and eliminates some of the spurious lattices discovered from randomly selected patches. Even though the window detection is automatic, the interaction here is purely to evaluate the grouping and serves to separate out the token extraction from the overall framework. MCMC grouping is then performed by considering the dominant peaks (or MSER features) as tokens. Fig. 5 (bottom row) show the inferred lattice configuration for a few NRT images. Note that only the generic lattice functions in (2) and (3) have been used here. A suitable image-based $\mathcal{B}$ function could have prevented some of the incorrect links between unlike elements. On images undergoing significant non-rigid distortions, correlation may also not produce a strong enough peak, causing some interior texels to be suppressed. Being less relevant to the current theme on buildings, we do not explicitly address these issues in this work.

3 Facade Analysis and Manipulation

The window grid provides vital cues about positioning, layout, and scale. We now give a brief overview of some simple techniques for extracting additional properties . To ease processing, we work with rectified images in this section.

3.1 Lattice Completion

Occlusions or errors in the rectangle hypotheses can cause missing nodes in the lattice. The already grouped elements facilitate parameter inference of the regular grid. The median height, width, and magnitudes of the horizontal and vertical $\mathbf{t}$ vectors are computed first. We then pick the node with the highest



(a) (b)

Fig. 6. Discovered grid shown as blue rectangles. Images were automatically rectified as a pre-processing step. Rectangles plotted in red are occluded or missing windows inferred from the result of grouping.

likelihood according to (1) as an origin. This completely specifies a regular grid that can be overlaid on the image to hypothesize missing or occluded lattice elements. When window dimensions deviate from the perfect grid assumption, we observed that windows on the same floor are similar and centered horizontally and vertically with its neighbors. After overlaying the regular grid on the image, each location is tested for detected windows. If the test fails, a missing grid element is inferred at the location aligned with its neighbors in the two principal directions. Figure 6 shows examples where the discovered grid (drawn in blue) is used to identify occluded grid elements (drawn in red).

3.2 Facade Segmentation

By making assumptions founded on common architectural trends, building pixels can be segmented out from a static image without any prior knowledge of appearance models. We first assume that the pixels immediately around the perimeter of a window belong to the building wall and exhibit a Gaussian distribution W. Although not valid for walls with multiple colors or shadows, many buildings do exhibit such uniform color properties. Similar to MRF-based segmentation [31], we assume that a color Gaussian Mixture Model (GMM) can describe the majority of pixels in the image. The RGB values are clustered into N (typically less than 10) distributions $\mathcal{G}_i : i \in 1..N$. Based on the homogeneous texture assumption, the mean and covariance of the distributions can be used to compute the Maximum Likelihood cluster $\mathcal{G}_w$ that corresponds best to the wall color W. The left column of Figure 7 shows the mask separating out the complicated foreground for a couple of images. We can also use the mask to compute approximate facade boundaries (right column) by thresholding on adjacent row and column sum differences.

3.3 Foreground Removal

Texture discovery gives a set of subimages centered on the tile elements that should be very similar to one another. Appearance discrepancies arising from

(a) (b) (c) (d)

Fig. 7. (a) and (c): Mask of the wall pixels after maximum likelihood classification; (b) and (d): thresholding on the row and column sum differences between adjacent locations can be used to approximate the facade extent (blue rectangles)

material variations, spatial resolution, and non-planar features can be described with low-dimensional Gaussian models or blurring and shifting of patches. However, foreground elements or reflections are best treated as outliers of the building tile pixel mode. To first detect foreground, we look at pixel values in corresponding locations over all tiles under the assumption that the background is visible in a majority of them. A robust measure of spread, the median absolute deviation (MAD) [32], can be used to assess which pixel values vary enough across the tiles to arouse suspicion of foreground.[1] Unreliable pixels are identified by thresholding their MADs—these are so-called *MAD outlier* pixels.

An obvious approach to background reconstruction is to do spatial inpainting on pixels marked out according to the MAD criterion. Figure 1(c) shows the result of inpainting MAD-induced holes in Figure 1(a) with the method of [9] under the special case that the inpainting source patches are the same sizes as and perfectly aligned with the discovered tiles. In spite of this, the more complex the tile interior is, the less effective inpainting would be in avoiding geometric and photometric artifacts.

Another possibility suggested by the alignment of patches is to treat the problem as one of eigenimage reconstruction. Assuming that a Gaussian process describes inter-tile appearance variation fairly well, we can use principal components analysis (PCA) to model it (e.g., [33,18]). The intuition is that we want to take each occluded tile in which some background is visible and "project" it down onto a set of background-only bases in order to lessen the foreground influence. However, with some fraction of the tiles "polluted" by unknown foreground elements, robust PCA (RPCA) techniques [34,32] are required. In practice, these methods had problems with our data when there were too many outlier pixels in a tile. Sine the MAD mask identifies the outlier pixels well, we use another PCA variant called probabilistic PCA (PPCA) [35] which works when missing data is explicitly identified beforehand. As the percentage of pixels occluded in a given tile rises, however, the reliability of the reconstruction naturally deteriorates. We mitigate this issue by reconstructing only tiles that have $\leq 25\%$ outliers in them. All other tiles are treated as fully occluded and simply sampled *de novo*

[1] A scalar MAD value is obtained at each pixel by computing it separately for each color channel and summing.

(a) (b) (c) (d)

Fig. 8. Foreground removal by eigenimage reconstruction for tile aligned images (a) and (c)

from the learned PPCA basis. Figures 1(d) and 8 show results of using PPCA on images containing complicated foreground or reflections within windows.

4 Conclusion

We draw the analogy that building facades are often examples of Near-Regular Textures, and showed that discovering these textures could provide valuable insight into the rest of the facade. We introduce a novel MRF/MCMC approach to discover grid patterns from images. We then presented techniques that use a partially discovered grid to infer occluded windows, segment out wall texture, identify foreground pixels, and reconstruct the background – all from a single image.

All the components described in the paper have much scope for extension. Foremost among these, an extensive evaluation of our lattice discovery technique to more general NRTs is being done. An image-specific likelihood function is required to prevent false links that might be topologically correct. One shortcoming of our proposal function is the inability to hypothesize a new token during MCMC, which would require Reversible Jump MCMC. We have also had encouraging results in parsing window interiors to describe them with split grammars. The PCA approach to recover the background currently disallows tile non-regularity such as those in Fig. 7. However, the segmentation mask and window grid provide enough information to allow inpainting of foreground pixels and copying of whole windows to maintain coherency. It is also important to note that each of the techniques presented in Section 3 can be replaced with more sophisticated methods. Alternative approaches such as gradient-domain methods [6] could be used for texture replacement as long as the facade structure discovered by the grouping framework is not violated during synthesis. Future work includes comparing some of these methods based on the quality of the recovered texture map.

References

1. Debevec, P., Taylor, C., Malik, J.: Modeling and rendering architecture from photographs. In: SIGGRAPH (1996)
2. Teller, S., Antone, M., Bodnar, Z., Bosse, M., Coorg, S., Jethwa, M., Master, N.: Calibrated, registered images of an extended urban area. Int. J. Computer Vision 53, 93–107 (2003)

3. Dick, A., Torr, P., Cipolla, R.: Modelling and interpretation of architecture from several images. Int. J. Computer Vision 60(2) (November 2004)
4. van den Heuvel, F.: Automation in Architectural Photogrammetry; Line-Photogrammetry for the Reconstruction from Single and Multiple Images. PhD thesis, Delft University of Technology, Delft, The Netherlands (2003)
5. Früh, C., Zakhor, A.: An automated method for large-scale, ground-based city model acquisition. Int. J. Comput. Vision 60(1), 5–24 (2004)
6. Wilczkowiak, M., Brostow, G., Tordoff, B., Cipolla, R.: Hole filling through photomontage. In: Proc. British Machine Vision Conference (2005)
7. Korah, T., Rasmussen, C.: Spatiotemporal inpainting for recovering texture maps of occluded building facades. IEEE Transactions on Image Processing 16(9), 2262–2271 (2007)
8. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: SIGGRAPH, pp. 417–424 (2000)
9. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. IEEE Trans. Image Processing 13(9) (2004)
10. Sun, J., Yuan, L., Jia, J., Shum, H.Y.: Image completion with structure propagation. ACM Transactions on Graphics 24, 861–868 (2005)
11. Mayer, H., Reznik, S.: Building faade interpretation from image sequences. In: Proc. of the ISPRS Workshop CMRT 2005 - Object Extraction for 3D City Models, Road Databases and Traffic Monitoring - Concepts, Algorithms and Evaluation (2005)
12. Neal, R.M.: Probabilistic inference using markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto (1993)
13. Wonka, P., Wimmer, M., Sillion, F., Ribarsky, W.: Instant architecture. ACM Transactions on Graphics 22, 669–677 (2003)
14. Alegre, F., Dellaert, F.: A probabilistic approach to the semantic interpretation of building facades. In: International Workshop on Vision Techniques Applied to the Rehabilitation of City Centres (2004)
15. Mueller, P., Zeng, G., Wonka, P., Gool, L.V.: Image-based procedural modeling of facades. In: Proceedings of ACM SIGGRAPH 2007. ACM Press, New York (2007)
16. Schaffalitzky, F., Zisserman, A.: Geometric grouping of repeated elements within images. In: Proceedings of the 9th British Machine Vision Conference, Southampton (1998)
17. Turina, A., Tuytelaars, T., Gool, L.V.: Efficient grouping under perspective skew. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2001)
18. Liu, Y., Lin, W.C., Hays, J.H.: Near regular texture analysis and manipulation. In: ACM Transactions on Graphics (SIGGRAPH 2004), vol. 23(3), pp. 368–376 (August 2004)
19. Liu, Y., Collins, R., Tsin, Y.: A computational model for periodic pattern perception based on frieze and wallpaper groups. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(3), 354–371 (2004)
20. Hays, J.H., Leordeanu, M., Efros, A.A., Liu, Y.: Discovering texture regularity as a higher-order correspondence problem. In: 9th European Conference on Computer Vision (May 2006)
21. Ahuja, N., Todorovic, S.: Extracting texels in 2.1d natural textures. In: Proc. IEEE Int. Conf. Computer Vision (ICCV 2007) (2007)
22. Li, S.Z.: Markov random field modeling in computer vision. Springer, Heidelberg (1995)
23. Hartelius, K., Carstensen, J.M.: Bayesian grid matching. IEEE Trans. Pattern Anal. Mach. Intell. 25(2), 162–173 (2003)

24. Lin, W.C., Liu, Y.: Tracking dynamic near-regular textures under occlusion and rapid movements. In: 9th European Conference on Computer Vision (2006)
25. Matsuyama, T., Miura, S., Nagao, M.: A structural analysis of natural textures by fourier transformation. In: Proc. Int. Conf. Pattern Recognition (1982)
26. Han, F., Zhu, S.C.: Bottom-up/top-down image parsing by attribute graph grammar. In: Proc. of the IEEE International Conference on Computer Vision (ICCV 2005) (2005)
27. Khan, Z., Balch, T., Dellaert, F.: Mcmc-based particle filtering for tracking a variable number of interacting targets. Pattern Analysis and Machine Intelligence 27(11), 1805–1918 (2005)
28. Hastings, W.: Monte carlo sampling methods using markov chains and their applications. Biometrika 57(1), 97–109 (1970)
29. Lee, S., Liu, Y.: Psu Near-Regular Texture Database (2007), http://vivid.cse.psu.edu/texturedb/gallery/
30. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proceedings of the British Machine Vision Conference (2002)
31. Rother, C., Kolmogorov, V., Blake, A.: Grabcut - interactive foreground extraction using iterated graph cuts. In: SIGGRAPH (2004)
32. la Torre, F.D., Black, M.: A framework for robust subspace learning. Int. J. Computer Vision 54, 117–142 (2003)
33. Turk, M., Pentland, A.: Face recognition using eigenfaces. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (1991)
34. Xu, L., Yuille, A.: Robust principal component analysis by self-organizing rules based on statistical physics approach. IEEE Transactions on Neural Networks 6(1), 131–143 (1995)
35. Roweis, S.: EM algorithms for PCA and SPCA. In: Advances in Neural Information Processing Systems (1997)

Nonrigid Image Registration Using *Dynamic* Higher-Order MRF Model

Dongjin Kwon[1], Kyong Joon Lee[1], Il Dong Yun[2,⋆], and Sang Uk Lee[1]

[1] School of EECS, Seoul Nat'l Univ., Seoul, 151-742, Korea
[2] School of EIE, Hankuk Univ. of F. S., Yongin, 449-791, Korea
{djk,kjoon}@cvl.snu.ac.kr, yun@hufs.ac.kr, sanguk@ipl.snu.ac.kr

Abstract. In this paper, we propose a nonrigid registration method using the Markov Random Field (MRF) model with a higher-order spatial prior. The registration is designed as finding a set of discrete displacement vectors on a deformable mesh, using the energy model defined by label sets relating to these vectors. This work provides two main ideas to improve the reliability and accuracy of the registration. First, we propose a new energy model which adopts a higher-order spatial prior for the smoothness cost. This model improves limitations of pairwise spatial priors which cannot fully incorporate the natural smoothness of deformations. Next we introduce a *dynamic* energy model to generate optimal displacements. This model works iteratively with optimal data cost while the spatial prior preserve the smoothness cost of previous iteration. For optimization, we convert the proposed model to pairwise MRF model to apply the tree-reweighted message passing (TRW). Concerning the complexity, we apply the *decomposed* scheme to reduce the label dimension of the proposed model and incorporate the linear constrained node (LCN) technique for efficient message passings. In experiments, we demonstrate the competitive performance of the proposed model compared with previous models, presenting both quantitative and qualitative analysis.

1 Introduction

Nonrigid image registration is the process of determining the geometric transformation between two images[1] which are not related by simple rigid or affine transforms. Although this process has been an essential stage in many computer vision applications, it is still one of the most challenging problem. In the last decade, many relevant techniques are proposed [1]. They can be distinguished as feature-based and image-based schemes. The feature-based scheme uses point landmarks extracted manually or automatically, and generates a deformation pattern based on the surface interpolation techniques which use matching pairs of landmarks [2,3,4]. The image-based scheme uses pixel-wise similarity measures

⋆ This work was supported by the Korea Science and Engineering Foundation(KOSEF) grant funded by the Korea government(MOST) (R01-2007-000-11425-0).

[1] We refer a fixed image as *reference* and a moving image as *input* during registration.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 373–386, 2008.

for comparing two images, and finds optimal deformation patterns by transforming the input image [5]. For both schemes, energy minimization approaches are conventionally applied to find optimal transformations. The energy is defined using similarity measures between landmarks or images and deformation energy of the surface. In general numerical methods based on the gradient descent have been used for minimizing this energy.

Recently, nonrigid registration methods [6,7] incorporating the state-of-the-art discrete energy optimization [8,9] come into the spotlight. These methods model the deformation pattern as a discrete label set where labels correspond to displacements of control points (nodes) placed on the square mesh. The energy is constructed using the standard pairwise MRF model as follows

$$E(x|\theta) = \sum\nolimits_{s\in\mathcal{V}} \theta_s(x_s) + \sum\nolimits_{(s,t)\in\mathcal{E}} \theta_{st}(x_s, x_t) \tag{1}$$

where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of edges incorporating neighborhood information of nodes, and x_s is the label of $s \in \mathcal{V}$. In this model, the data cost θ_s is computed using similarity measures between reference and input images and the smoothness cost θ_{st} is computed using displacement differences between s and t. Glocker *et al.* [6] use the weighted block matching cost for θ_s where the free-form deformation (FFD) model [10] is used for weighting coefficient. The FFD model is also used for generating a pixel-wise deformation field from discrete displacements. For θ_{st}, they use the truncated pairwise spatial prior as follows

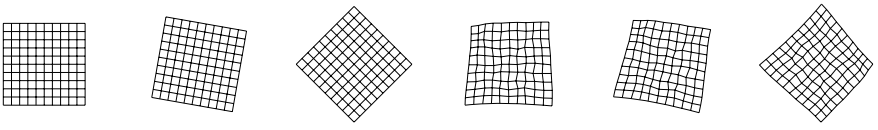
$$\theta_{st}(x_s, x_t) = \lambda_{st} \min\left(\|\mathbf{d}(x_s) - \mathbf{d}(x_t)\|, T\right) \tag{2}$$

where λ_{st} is the regularization constant, $\mathbf{d}(x_s)$ represents the displacement vector corresponding to the label x_s, and T is a threshold for truncation. In the paper, their method produces better quality than the state of the art [5] with almost 60 times faster speed for 3D medical images. Shekhovtsov *et al.* [7] use the uniformly weighted block matching cost for θ_s. For θ_{st}, the modified linear pairwise spatial prior is used as follows

$$\theta_{st}(x_s, x_t) = \lambda_{st} \sum\nolimits_i \max\left(c_1\left(|d_i(x_s) - d_i(x_t)| - 1\right), c_2|d_i(x_s) - d_i(x_t)|\right) \tag{3}$$

where $d_i(x_s)$ is i^{th} coordinate value of displacement vector $\mathbf{d}(x_s)$. As usually $c_1 \gg c_2$ is used, this prior assigns almost zero cost to small displacement difference between neighborhoods. Compared to (2), small movements of nodes are allowed more freely. In the paper, their method produces promising results although they did not provide any comparative evaluations.

However, the energy models using (2) or (3) have limitations. The pairwise potentials (2) and (3) penalize the global transformations of the mesh such as rotation and scaling movements. Mesh nodes must be remained at initial position or translated all together to get low energy. Although (3) relaxed this restriction by assigning low penalty in a small range of $|d_i(x_s) - d_i(x_t)|$, this is still not the natural representation of smoothness energy of the mesh. In Fig. 1, we show that these pairwise potentials can not preserve smoothness energy for simple global



(a) 0° (b) 10° (c) 135° (d) 0° (e) 10° (f) 135°

Spatial Prior	(a)	(b)	(c)	(d)	(e)	(f)
Pairwise (2)	0.0	1.2×10^3	2.2×10^3	8.4×10^2	1.4×10^3	2.2×10^3
Pairwise (3)	0.0	1.0×10^3	1.7×10^4	7.0×10^2	1.4×10^3	1.7×10^4
Higher-Order (7)	0.0	≈ 0.0	≈ 0.0	4.2×10^3	4.1×10^3	4.6×10^3

Fig. 1. The smoothness energy on synthetic meshes. {(b),(c)} are generated from a 11×11 square mesh (a) (grid spacing = 32) by rotating 10° and 135°. Similarly, {(e),(f)} are generated from (d) which is deformed synthetically from (a). One can see the pairwise priors produce largely different smoothness energies for each transformation, while the higher-order prior preserves almost same energy. (We use $\lambda_{st} = \lambda_{stu} = 1$, $T = 10$, $c_1 = 1$, and $c_2 = 10^{-3}$ for calculating potentials).

transformations. The other limitation is the energy model using (2) or (3) can not guarantee the reliability between iterative registrations. Using conventional block matching scheme, the data cost θ_s in (1) is not an optimal measurement. To improve registration quality, we execute registration sequentially using intermediate registered input images. If an independent energy model with a new mesh is used for each registration, errors on displacements are normally accumulated.

In this paper we propose a nonrigid registration method using a new MRF based energy model which improves above limitations. The proposed energy model incorporates the higher-order spatial interactions to represent natural smoothness of the mesh. In the proposed model, we use a deformable mesh applying the FFD model based on B-splines for representing the nonrigid transformation where the FFD is successfully applied in nonrigid image registration [5,6]. To generate optimal registration results, we introduce the *dynamic* version of our MRF energy model. The dynamic energy model is used to control the spatial priors for successive registrations. We propose an efficient energy optimization method on the higher-order MRF model which is based on the tree reweighted message passing (TRW) [11,8] method.

2 Preliminaries

Notations. Let $\mathcal{G}_{\mathcal{E}} = (\mathcal{V}, \mathcal{E})$ be an undirected graph where $\mathcal{V}$ and $\mathcal{E}$ are the set of nodes and edges, respectively, and $\mathcal{G}_{\mathcal{F}} = (\mathcal{V}, \mathcal{F})$ be a factor graph [12] with the set of factors $\mathcal{F}$. For each $s \in \mathcal{V}$, let x_s be a label taking values in some discrete set $\mathcal{L}$. If we define a function $\mathbf{d} : \mathcal{L} \rightarrow \mathbb{R}^n$ for mapping labels to n-dimensional displacements, each label x_s corresponds to a displacement vector $\mathbf{d}(x_s)$. For conveniences, we assume each dimension of displacements has same discrete set of values $\mathcal{D} = \{-D, -D+1, \ldots, 0, \ldots, D-1, D\}$ where $D \in \mathbb{N}$

controls a displacement width[2]. For a set $\mathcal{D}$, we assign a corresponding label set $l = \{l_1, l_2, \ldots, l_K\}$, then $|\mathcal{L}| = |\mathcal{D}^n| = (2D+1)^n = K^n$.

In $\mathcal{G}_\mathcal{E}$, a unary potential $\theta_s(x_s)$ is defined for each node $s \in \mathcal{V}$ and a pairwise potential $\theta_{st}(x_s, x_t)$ is defined for each edge $(s,t) \in \mathcal{E}$ where $t \in \mathcal{V}$. In $\mathcal{G}_\mathcal{F}$, a higher-order potential $\theta_a(\mathbf{x}_a)$ is defined for each factor $a \in \mathcal{F}$ where $\mathbf{x}_a$ is a label vector of nodes connected to a, we also use θ_s and θ_{st} notations if a factor has only one or two nodes. If a factor a connects nodes s, t, and u, a ternary potential $\theta_{stu}(x_s, x_t, x_u)$ is defined. In this case, we denote $a\{stu\}$ for a factor a to emphasis node to factor connections. We also use $(s,t,u) \in \mathcal{F}$ to represent $a\{stu\} \in \mathcal{F}$ when we do not need to use factor representations explicitly.

Deformable Mesh. For an input image, we construct a set $\mathcal{V}$ which consists of nodes v placed in a grid with a spacing δ. If we denote the domain of the input image as $\Omega = \{(x,y)|0 \le x < X, 0 \le y < Y\}$, the size of image covered by $\mathcal{V}$ is $(X+2\delta) \times (Y+2\delta)$. For a given $\mathcal{V}$, a displacement vector $\mathbf{d}(\mathbf{x})$ of an image pixel $\mathbf{x} \in \Omega$ is represented by neighboring displacement vectors $\mathbf{d}(v)$ of nodes v using conventional FFD model based on cubic B-splines [10,5] as follows

$$\mathbf{d}(\mathbf{x}) = \sum_{l=0}^{3} \sum_{m=0}^{3} B_l(a) B_m(b) \mathbf{d}(v_{i+l,j+m}) \tag{4}$$

where $i = \lfloor x/\delta \rfloor + 1$, $j = \lfloor y/\delta \rfloor + 1$, $a = x/\delta - \lfloor x/\delta \rfloor$, $b = y/\delta - \lfloor y/\delta \rfloor$, and B_l is the lth basis function of the uniform cubic B-spline[3].

For a given nodes $\mathcal{V}$, a graph $\mathcal{G}_\mathcal{E}$ or $\mathcal{G}_\mathcal{F}$ is constructed after the pairwise or higher-order interactions between nodes is determined. When we transform the locations of each node, a transformed image is interpolated using (4).

Data Cost Computation. The unary term $\theta_s(x_s)$ of (1) which measures the cost when a node s has a label x_s is defined as

$$\theta_s(x_s) = f(s_x, s_y, s_x + d_x(x_s), s_y + d_y(x_s)) \; .$$

This cost is calculated by the dissimilarity measure f using two image patches centered on (s_x, s_y) in a reference image and $(s_x + d_x(x_s), s_y + d_y(x_s))$ in a input image, respectively.

As this representation is flexible, we can incorporate various dissimilarity measures to f. In this paper we use the normalized cross correlation (NCC) measure as it gives the best results comparing with other measures in intra-modal data set [6].

3 MRF Energy Model with Higher-Order Spatial Priors

In the literature, deformation energy E_D of mesh model is usually defined as a sum of squared second derivatives of mesh nodes [13]. E_D describes the natural

[2] For an example of 2-dimensional displacements, $\mathbf{d}(x_s) = d_x(x_s), d_y(x_s) \in \mathcal{D}^2$ where $d_x(x_s) \in \mathcal{D}$ and $d_y(x_s) \in \mathcal{D}$.

[3] $B_0(t) = (1-t)^3/6$, $B_1(t) = (3t^3 - 6t^2 + 4)/6$, $B_2(t) = (-3t^3 + 3t^2 + 3t + 1)/6$, $B_3(t) = t^3/6$.

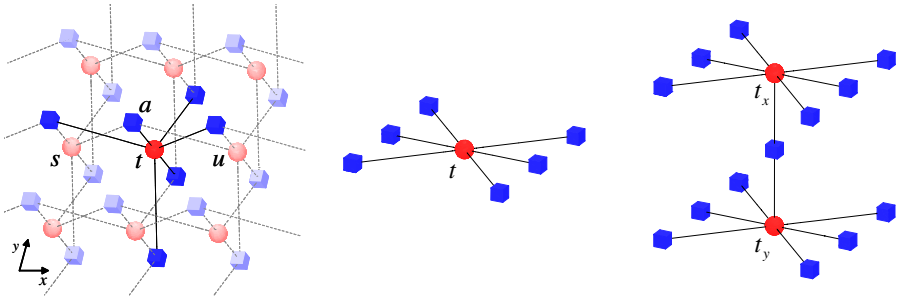


Fig. 2. The proposed higher-order MRF models. Nodes and factors are represented as red spheres and blue cubes respectively. (a) shows a 3×3 graph model for (6). A factor $a \in \mathcal{F}$ connects collinear (in x-direction) nodes $\{s, t, u\} \in \mathcal{V}$. (b) shows node t with connected factors which correspond highlighted factors in (a). (c) shows a corresponding part of (b) for the *decomposed* model (11). This model copies graph structure (b) to each coordinate layers $\mathcal{V}^x$ and $\mathcal{V}^y$, and adds pairwise factors $\mathcal{F}^{xy}$ which connect nodes (t_x and t_y) belonged to the same node (t) in (b).

representation of inherent deformedness of the mesh which depends only on the relative locations of mesh nodes:

$$E_D \propto \int_D \left(\left\| \partial^2 \mathcal{T} / \partial x^2 \right\|^2 + \left\| \partial^2 \mathcal{T} / \partial y^2 \right\|^2 \right) d\mathbf{x}$$

where $\mathcal{T}$ is a transformation function defined on image pixels $\mathbf{x} \in \Omega$. We approximate E_D as a parameterized form of mesh nodes [3,4]:

$$E_D \approx \sum\nolimits_{(i,j,k) \in L} (-x_i + 2x_j - x_k)^2 + (-y_i + 2y_j - y_k)^2 \tag{5}$$

where L is an index set of successive collinear three nodes and (x_i, y_i) is a coordinate of a node having index i. As this spatial prior related more than two nodes, the pairwise MRF energy model (2) and (3) can not be integrated.

To incorporate this natural representation of mesh deformation energy to the graph model, we propose a new MRF energy model with higher-order spatial priors described as follows

$$E(x|\theta) = \sum_{s \in \mathcal{V}} \theta_s(x_s) + \sum_{(s,t,u) \in \mathcal{F}} \theta_{stu}(x_s, x_t, x_u) \tag{6}$$

where $\mathcal{F}$ is a set of collinear three nodes corresponding to L. The higher-order spatial prior $\theta_{stu}(x_s, x_t, x_u)$ is defined as

$$\theta_{stu}(x_s, x_t, x_u) = \lambda_{stu} \Big\{ g\big(d_x(x_s), d_x(x_t), d_x(x_u)\big) + g\big(d_y(x_s), d_y(x_t), d_y(x_u)\big) \Big\} \tag{7}$$

where $g(a, b, c) = (-a + 2b - c)^2$. This spatial prior produces exactly same deformation energy with (5). In Fig. 1, we show that the proposed potential (7)

preserve smoothness energy for simple global transformations while pairwise potentials (2) and (3) can not. We illustrate the proposed graph model $\mathcal{G}_{\mathcal{F}} = (\mathcal{V}, \mathcal{F})$ in Fig. 2(a) and 2(b).

3.1 *Dynamic* Higher-Order MRF Energy Model

As we described in Section 1, we can execute registration iteratively using intermediate registered input images to improve registration quality. However if independent energy model is used between sequential registrations, registration errors are normally accumulated. To overcome this limitation we use *dynamic*[4] MRF energy model which recycles the mesh and the smoothness energy of previous registration step. In this model, mis-registered regions can have the opportunity to fix with intermediate input images while the mesh deformation is continuously controlled by the smoothness energy.

If we denote $\mathcal{T}^T$ as a transformed location of mesh nodes at time T, $\mathcal{T}^T$ is represented as the sum of mesh nodes at time $T-1$ and transform vectors at time T:

$$\mathcal{T}^T(\mathbf{x}) = \mathcal{T}^{T-1}(\mathbf{x}) + \mathbf{d}(\mathbf{x}^T) \; .$$

Then the *dynamic* higher-order energy model is defined as follows

$$E(x^T|\theta^T, \mathcal{T}^{T-1}(\mathbf{x})) = \sum_{s\in\mathcal{V}} \theta_s^T(x_s^T) + \sum_{(s,t,u)\in\mathcal{F}} \theta_{stu}\Big(x_s^T, x_t^T, x_u^T|\mathcal{T}^{T-1}(\mathbf{x})\Big)$$

where the unary potential θ^T is calculated from the registered input image at $T-1$ and the reference image, and the spatial prior is defined as follows

$$\begin{aligned}\theta_{stu}(x_s, x_t, x_u|\mathcal{T}(\mathbf{x})) = \lambda_{stu}\Big\{ & g\big(\mathcal{T}_x(x_s)+d_x(x_s), \mathcal{T}_x(x_t)+d_x(x_t), \mathcal{T}_x(x_u)+d_x(x_u)\big) \\ & + g\big(\mathcal{T}_y(x_s) + d_y(x_s), \mathcal{T}_y(x_t) + d_y(x_t), \mathcal{T}_y(x_u) + d_y(x_u)\big)\Big\} \; .\end{aligned}$$

As $\theta_{stu}(x_s, x_t, x_u|\mathcal{T}^{T-1}(\mathbf{x}))$ depends on the mesh location at $T-1$, it retains the smoothness energy at $T-1$.

3.2 Efficient MRF Energy Model with Decomposed Scheme

In [7], an efficient energy modeling method named *decomposed* scheme is introduced. This scheme divides x and y displacements by making two layers of nodes $\mathcal{V}^x$ and $\mathcal{V}^y$ from the original nodes $\mathcal{V}$. The edge sets include $\mathcal{E}^x$ and $\mathcal{E}^y$ for intra-layer interaction potentials and $\mathcal{E}^{xy}$ for inter-layer interaction potentials. Note that the graph $\mathcal{G}^x = (\mathcal{V}^x, \mathcal{E}^x)$ and $\mathcal{G}^y = (\mathcal{V}^y, \mathcal{E}^y)$ have the identical structures with the original graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and the inter-layer edges $\mathcal{E}^{xy}$ link two nodes

[4] We refer to MRF models varying over iterations as *dynamic*. This term is firstly introduced to the vision community in [14]. However our *dynamic* MRF models are focused on reusing intermediate solutions while maintaining smoothness energies in the registration perspective.

$v^x \in \mathcal{V}^x$ and $v^y \in \mathcal{V}^y$ which correspond to the same node $v \in \mathcal{V}$ in the original graph. Then the energy model converted from (1) is following[5]

$$E(x|\theta) = \sum_{(s,t)\in\mathcal{E}^x\cup\mathcal{E}^y} \theta_{st}(x_s, x_t) + \sum_{(s,t)\in\mathcal{E}^{xy}} \theta_{st}(x_s, x_t) \tag{8}$$

where $\theta_{st}(x_s, x_t) = f(s_x, t_y, s_x + d(x_s), t_y + d(x_t))$ for $(s,t) \in \mathcal{E}^{xy}$ and following potential is used for $(s,t) \in \mathcal{E}^x \cup \mathcal{E}^y$:

$$\theta_{st}(x_s, x_t) = \lambda_{st} \max\left(c_1(|d(x_s) - d(x_t)| - 1), c_2|d(x_s) - d(x_t)|\right) . \tag{9}$$

We found this scheme can be extended to the general energy model if the label corresponds to multi-dimensional vector such as displacement, and the smoothness cost is separated into each dimensional term. We can use the pairwise prior (2) if we only change (9) as following form

$$\theta_{st}(x_s, x_t) = \lambda_{st} \min\left(|d(x_s) - d(x_t)|, T\right) \tag{10}$$

where we assume that $\|\cdot\|$ in (2) is 1-norm. As the higher-order spatial prior (7) is also decomposed into each dimension, we apply the decomposed scheme to the proposed energy model (6) as follows

$$E(x|\theta) = \sum_{(s,t)\in\mathcal{F}^{xy}} \theta_{st}(x_s, x_t) + \sum_{(s,t,u)\in\mathcal{F}^x\cup\mathcal{F}^y} \theta_{stu}(x_s, x_t, x_u) \tag{11}$$

where the potentials are defined as

$$\theta_{st}(x_s, x_t) = f(s_x, t_y, s_x + d(x_s), t_y + d(x_t)) ,$$
$$\theta_{stu}(x_s, x_t, x_u) = \lambda_{stu} g\left(d(x_s), d(x_t), d(x_u)\right) .$$

The ternary potential $\theta_{stu}(x_s, x_t, x_u)$ is decomposed form of (7). We describe the proposed graph model $\mathcal{G}_{\mathcal{F}} = (\mathcal{V}^x \cup \mathcal{V}^y, \mathcal{F}^x \cup \mathcal{F}^y \cup \mathcal{F}^{xy})$ in Fig. 2(c). Note that every decomposed energy model produces exactly same energy with the original model for the same label configuration.

4 Max-Product Belief Propagation on Factor Graphs

To optimize the proposed energy model, the max-product belief propagation (BP) can be used. The message update equations for (11) are as follows

$$m_{a\{stu\}\to s}(x_s) = \min_{x_t, x_u}\Big\{\theta_{stu}(x_s, x_t, x_u) + \sum_{c\in N(t)\setminus a} m_{c\to t}(x_t) + \sum_{c\in N(u)\setminus a} m_{c\to u}(x_u)\Big\},$$
$$m_{a\{st\}\to s}(x_s) = \min_{x_t}\Big\{\theta_{st}(x_s, x_t) + \sum_{c\in N(t)\setminus a} m_{c\to t}(x_t)\Big\}$$

where $N(t)$ denotes a set of neighboring factors for node t. In this representation, message update equations only incorporate factor to node messages [15]. The first equation updates intra-layer messages where $(s,t,u) \in \mathcal{F}^x \cup \mathcal{F}^y$ and second equation updates inter-layer messages where $(s,t) \in \mathcal{F}^{xy}$.

[5] We do not describe the unary potential $\theta_s(x_s)$ ($s \in \mathcal{V}^x \cup \mathcal{V}^y$) which is a zero vector though it is used in the reparameterization stage of TRW [8].

4.1 Efficient Message Update Using Linear Constraint Nodes

As $g(a,b,c) = \big((a,b,c)\cdot(-1,2,-1)\big)^2$ in (7), the proposed higher-order spatial prior θ_{stu} (including *dynamic* version) satisfies the linear constraint node (LCN) conditions [16]. Therefore we can calculate the message update equations more efficiently using the LCN techniques. Let us define following notations

$$M_t(x_t) = \sum\nolimits_{c\in N(t)\setminus a} m_{c\to t}(x_t)\ ,$$

$$\theta^s_{stu}(x_s,y) = \theta_{stu}\big(x_s, d^{-1}\big((y+d(x'_u))/2\big), x'_u\big) \tag{12}$$

where $d^{-1}(y)\in\mathcal{L}$ is a label which corresponds to y and x'_u is any discrete value satisfying $\{x'_u\in\mathcal{L}\}\wedge\{d^{-1}\big((y+d(x'_u))/2\big)\in\mathcal{L}\}$. Then an efficient message update equation using the LCN is described as

$$m_{a\{stu\}\to s}(x_s) = \min_y\Big\{\theta^s_{stu}(x_s,y) + \min_{x_u}\Big(M_t\Big(d^{-1}\big((y+d(x_u))/2\big)\Big) + M_u(x_u)\Big)\Big\}$$

where the calculation for x_u is skipped when $d^{-1}\big((y+x_u)/2\big)\notin\mathcal{L}$. The equations for $m_{a\{stu\}\to t}(x_t)$ and $m_{a\{stu\}\to u}(x_u)$ can be similarly described. This equations produce exactly same results comparing to the original equations, however the time complexity for updating messages is shortened from $\mathcal{O}(|\mathcal{L}|^3)$ to $\mathcal{O}(|\mathcal{L}|^2)$.

5 Tree-Reweighted Message Passing on Factor Graphs

The major problem of BP on higher-order factor graphs is that message update scheduling is not easy and the convergence is not guaranteed. We apply TRW message passing [11,8] which provide the lower bound guaranteed not to decrease. Recent comparative study shows the TRW gives the state-of-the-art performances among the various discrete optimization methods [17].

5.1 Conversion from Factor Graphs to Pairwise Interactions

As the TRW theory is built on the pairwise MRF, we need to convert factor graphs representations to pairwise interactions to use TRW. We apply the conventional conversion procedure described in [18,11] and introduce its efficient implementation. Let us consider a factor $a\{stu\}$ which has a ternary potential $\theta_{stu}(x_s,x_t,x_u)$. The conversion is simply done by replacing factor nodes with auxiliary variable nodes. We associate a new label $z\in\mathcal{Z}$ with the auxiliary node which replaces the factor a where $\mathcal{Z}$ is Cartesian product of label spaces of connected three nodes ($\mathcal{Z}=\mathcal{L}_s\times\mathcal{L}_t\times\mathcal{L}_u$). Then unary and pairwise potentials of this converted energy model are described as follows

$$\psi_{ai}(z,x_i) = \begin{cases} 0 & \text{if } z_i = x_i \\ \infty & \text{otherwise} \end{cases} \quad \forall i\in\{s,t,u\}\ ,$$

$$\psi_a(z) = \theta_{stu}(z_s,z_t,z_u)\ , \quad \psi_s(x_s)=\psi_t(x_t)=\psi_u(x_u)=0$$

where any possible value z has one-to-one correspondence with a triplet (z_s, z_t, z_u) if $\{z_s, z_t, z_u\} \in \mathcal{L}$. The higher-order spatial prior in (6) can be converted to the sum of these pairwise interactions:

$$\theta_{stu}(x_s, x_t, x_u) = \psi_a(z) + \sum_{i \in \{s,t,u\}} \left(\psi_i(x_i) + \psi_{ai}(z, x_i)\right) . \tag{13}$$

Applying (13) to the proposed energy model with decomposed scheme (11) , the converted energy is represented as[6]

$$E(x|\theta,\psi) = \sum_{s \in \mathcal{V}_{\mathcal{O}}^x \cup \mathcal{V}_{\mathcal{O}}^y} \theta_s(x_s) + \sum_{(s,t) \in \mathcal{E}_{\mathcal{F}}^{xy}} \theta_{st}(x_s, x_t) + \sum_{a \in \mathcal{V}_{\mathcal{F}}^x \cup \mathcal{V}_{\mathcal{F}}^y} \psi_a(z) + \sum_{(a,i) \in \mathcal{E}_{\mathcal{F}}^x \cup \mathcal{E}_{\mathcal{F}}^y} \psi_{ai}(z, x_i) \tag{14}$$

where $\mathcal{V}_{\mathcal{O}}^x \cup \mathcal{V}_{\mathcal{O}}^y$ is the set of ordinary nodes, $\mathcal{V}_{\mathcal{F}}^x \cup \mathcal{V}_{\mathcal{F}}^y$ is the set of auxiliary nodes, $\mathcal{E}_{\mathcal{F}}^{xy}$ is the set of edges between ordinary nodes, and $\mathcal{E}_{\mathcal{F}}^x \cup \mathcal{E}_{\mathcal{F}}^y$ is the set of edges between ordinary nodes and auxiliary nodes. The original factor graph $\mathcal{G}_{\mathcal{F}} = (\mathcal{V}^x \cup \mathcal{V}^y, \mathcal{F}^x \cup \mathcal{F}^y \cup \mathcal{F}^{xy})$ is converted to the pairwise undirected graph $\mathcal{G}_{\mathcal{E}} = (\mathcal{V}_{\mathcal{O}}^x \cup \mathcal{V}_{\mathcal{O}}^y \cup \mathcal{V}_{\mathcal{F}}^x \cup \mathcal{V}_{\mathcal{F}}^y, \mathcal{E}_{\mathcal{F}}^x \cup \mathcal{E}_{\mathcal{F}}^y \cup \mathcal{E}_{\mathcal{F}}^{xy})$.

5.2 Efficient Implementation of TRW Algorithm

The basic procedure of the optimization procedure for the proposed model follows Kolmogorov's TRW-S implementation [8]. However, a direct implementation of TRW-S on converted energy equation (14) has a higher time complexity and requires a larger memory compared to the original equations. Each pairwise potential ψ_{ai} between an auxiliary node and an ordinary node need $\mathcal{O}(|\mathcal{L}|^4)$ dimensional space, and a message $m_{i \to a}$ from an ordinary node i to an auxiliary node a requires $\mathcal{O}(|\mathcal{L}|^3)$ dimensional space. These larger potentials and messages cause a higher time complexity.

Fortunately, the storage for ψ_{ai} can be neglected because it adds nothing to the messages in the max-product message passings. We are logically discard all summations when $z_i \neq x_i$. Moreover, we can modify $m_{i \to a}$ to have only $\mathcal{O}(|\mathcal{L}|)$ dimensional space, because $m_{i \to a}$ maps a summation over $\mathcal{O}(|\mathcal{L}|)$ dimensional messages $m_{b \to i}$ $(b \in N(i) \backslash a)$ to $\mathcal{O}(|\mathcal{L}|^3)$ dimensional space according to ψ_{ai}. This shortened message is reconstructed to the original during updating $m_{a \to i}$. The resulting message update equations between auxiliary node $a\{s,t,u\}$ and ordinary node s are following[7]

$$m_{s \to a}(x_s) = \gamma_{sa} \cdot \left(\theta_s(x_s) + \sum\nolimits_{b \in N(s)} m_{b \to s}(x_s)\right) - m_{a \to s}(x_s) ,$$

$$m_{a \to s}(x_s) = \min_{x_t, x_u} \left\{\gamma_{as} \cdot \left(\theta_{stu}(x_s, x_t, x_u) + \sum_{i \in N(a)} m_{i \to a}(x_s)\right) - m_{s \to a}(x_s)\right\}$$

[6] We expose the unary potential $\theta_s(x_s)$ to emphasis ordinary-auxiliary node relations. However this term is used in the reparameterization stage of TRW [8].

[7] We omit δ terms used for calculating the lower bound in these representations.

where $\gamma_{st} = 1/n_s$ and n_s is the number of trees containing node s [8]. Note that $m_{s\to a}$ is a shortened $\mathcal{O}(|\mathcal{L}|)$ message as described above, and we do not use ψ_{ai} explicitly. Message update equations between ordinary nodes are same to [8].

5.3 Message Update Using Linear Constraint Node

Using the LCN techniques [16] more efficient message passings for TRW algorithms is possible. Similar to BP on factor graphs in Section 4.1, message update equations for TRW using the LCN produce exactly same results with the original equations. The time complexity of a message passing for the auxiliary to ordinary node is reduced from $\mathcal{O}(|\mathcal{L}|^3)$ to $\mathcal{O}(|\mathcal{L}|^2)$. The message update equation for $m_{a\{stu\}\to s}$ is described as follows

$$Y^s(y) = \min_{x_u}\left\{m_{t\to a}\big(d^{-1}\big((y + d(x_u))/2\big)\big) + m_{u\to a}(x_u)\right\},$$
$$m_{a\to s}(x_s) = \gamma_{as}\cdot\min_y\left\{\theta^s_{stu}(x_s, y) + Y^s(y)\right\} + (\gamma_{as} - 1)\cdot m_{s\to a}(x_s)$$

where $\theta^s_{stu}(x_s, y)$ is defined in (12). The equations for $m_{a\to t}$ and $m_{a\to u}$ can be similarly described.

6 Experimental Results

In order to evaluate the proposed method, we use two types of data sets. One is a set of synthetically deformed images with ground truth deformations and the other is a set of real images with no ground truth information. We test three types of energy functions (parameters are empirically chosen):

- E^1_p: (8) with pairwise prior (10) using $\lambda_{st} = 10^{-4}$, $T = 10$
- E^2_p: (8) with pairwise prior (9) using $\lambda_{st} = 10^{-4}$, $c_1 = 1$, $c_2 = 10^{-3}$
- E_h: higher-order energy model (14) using $\lambda_{stu} = 6.4 \times 10^{-3}/\delta^2$

where λ_{stu} includes the mesh spacing δ to make θ_{stu} scale invariant. For data cost computation, we use the cubic B-spline weighted NCC using gray-scaled images. For energy minimization, the conventional TRW-S algorithm [8] is used for E^1_p and E^2_p, and the TRW-S using message update equations in Section 5 is used for E_h.

Seamless Integration with Feature-Based Registration. To cover wide displacement ranges in real environment, we need to use a large number of label K, however MRF optimization for large K will turn to be intractable problem. Moreover, rotation and scaling invariant data cost computation is another issue. Therefore we integrate the feature-based registration which covers large global transformations in the *dynamic* energy model framework. In experiments, we take a simple strategy: if we denote initial mesh position as $\mathcal{T}^0$, a feature-based registration which uses SIFT features with assuming perspective transformation [19] is used for generating $\mathcal{T}^1$ from $\mathcal{T}^0$. $\{\mathcal{T}^T | T \geq 2\}$ is generated from $\mathcal{T}^{T-1}$ using dynamic energy model for E_h. However we do not apply dynamic model

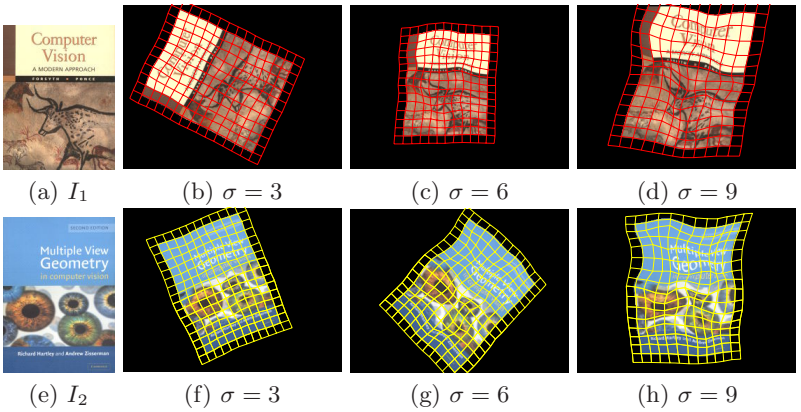
Table 1. Average displacement errors on synthetic data sets

\ Data Set			$I_1(\sigma=3)$		$I_1(\sigma=6)$		$I_1(\sigma=9)$		$I_2(\sigma=3)$		$I_2(\sigma=6)$		$I_2(\sigma=9)$	
E	δ	$\mathcal{T}$	d_{rms}	d_{max}	d_{rms}	d_{max}	d_{rms}	d_{max}	d_{rms}	d_{max}	d_{rms}	d_{max}	d_{rms}	d_{max}
E_p^1	16	$\mathcal{T}^2$	0.92	2.90	2.55	10.78	6.05	22.29	1.03	**3.35**	2.22	7.22	4.43	16.93
E_p^1	16	$\mathcal{T}^4$	0.82	2.79	1.84	8.76	4.81	22.59	1.00	4.65	1.79	7.62	3.72	15.85
E_p^2	16	$\mathcal{T}^2$	0.90	3.18	2.47	10.62	5.77	21.08	1.02	3.41	2.21	7.58	4.33	15.78
E_p^2	16	$\mathcal{T}^4$	0.81	3.20	1.72	7.85	4.54	21.71	0.93	4.09	1.81	7.76	3.46	14.96
E_h	16	$\mathcal{T}^2$	0.90	2.72	2.51	10.77	5.50	20.20	1.01	3.55	2.19	7.68	4.37	16.08
E_h	16	$\mathcal{T}^4$	**0.79**	**2.65**	**1.67**	**6.49**	**3.47**	**13.59**	**0.91**	3.61	**1.62**	**5.94**	**3.43**	**14.10**
E_p^1	8	$\mathcal{T}^2$	0.68	2.80	1.98	10.38	5.31	21.99	0.75	**2.85**	1.50	5.69	3.76	16.28
E_p^1	8	$\mathcal{T}^4$	0.72	4.04	1.44	9.43	4.08	20.08	0.83	5.34	1.13	5.59	2.81	14.19
E_p^2	8	$\mathcal{T}^2$	0.66	2.98	1.82	9.96	5.30	21.43	**0.71**	3.42	1.44	6.32	3.66	16.78
E_p^2	8	$\mathcal{T}^4$	0.68	4.06	1.31	9.20	3.90	18.76	0.88	6.04	1.14	6.06	2.72	15.74
E_h	8	$\mathcal{T}^2$	0.69	**2.73**	1.94	10.01	5.36	21.58	0.74	2.86	1.60	6.62	3.85	17.75
E_h	8	$\mathcal{T}^4$	**0.65**	2.89	**1.00**	**5.57**	**3.22**	**17.04**	0.73	3.93	**1.04**	**5.57**	**2.40**	**13.68**

for E_p^1 and E_p^2 as they perform poorly on the various mesh initializations by $\mathcal{T}^1$. For fair comparisons with E_h, we use spatial priors which is independent to time $T-1$ while data cost is updated on each time T for generating $\{\mathcal{T}^T | T \geq 2\}$ in E_p^1 and E_p^2.

6.1 Registration Using Synthetically Deformed Data

Given a base image, we generate a synthetically deformed data set by following steps: 1. Pick a point set P_1 placed on square grid with spacing $\delta_0 = 30$, 2. Perturb point locations with random variation $[-\sigma, \sigma]$, 3. Transfer points with random rotation, translation, and scaling $[0.7, 1.0]$ (let this transformed point set as P_2), 4. Interpolate deformation pattern using thin-plate spline (TPS) [13]



(a) I_1 (b) $\sigma = 3$ (c) $\sigma = 6$ (d) $\sigma = 9$

(e) I_2 (f) $\sigma = 3$ (g) $\sigma = 6$ (h) $\sigma = 9$

Fig. 3. Selected images from synthetic data set. (b-d) and (f-h) are synthetic images generated using base image (a) and (e) respectively (TPS meshes are overlaid).

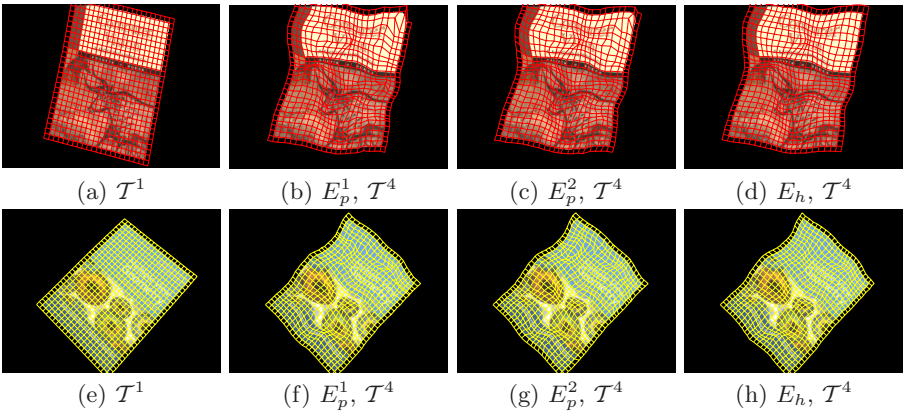


Fig. 4. Selected results from synthetic data set. (a-d) and (e-h) are registration results using (d) and (g) in Fig. 3 as reference images. {(a),(e)} are alignment results. {(b),(f)}, {(c),(g)}, and {(d),(h)} are generated using E_p^1, E_p^2, and E_h respectively.

using P_1 and P_2 correspondences. This data sets reflect wide displacement ranges in real environments. For each base image I_1 (size:160×208) and I_2 (size: 160×224), 10 synthetic images (size:320×240) are generated for each $\sigma = 3$, 6, and 9. We show some images from synthetic data set in Fig. 3.

The registration is performed between a image from synthetic data set (reference) and a corresponding base image (input) using E_p^1, E_p^2, E_h with $\delta = 8, 16$ and $D = 8$ ($\mathbf{d} \in [-8, 8]^2$,$K = 17^2$). We show average displacements errors in Table 1 where d_{rms} and d_{max} are the root mean square error (RMSE) and the maximum of the lengths of displacement difference vectors between the ground truth and the registration result. One can see the performances of the proposed energy model E_h in $\mathcal{T}^4$ are superior in almost every cases. (We indicate the best result as bold face numbers in each data set.) The performance difference is greater in larger σ data sets. In general, results on $\mathcal{T}^4$ are better than $\mathcal{T}^2$, and d_{max} at $\mathcal{T}^4$ is smaller in E_h than E_p^1 or E_p^2. In Fig.4, one can see the proposed model E_h produces more accurate and reliable result than E_p^1 or E_p^2.

6.2 Registration Using Real Data

The real data set includes photos (size:320×240) capturing real object deformations. We use T-shirts printed with base images I_1 and I_2 in Section 6.1. The registration is performed between an image from real data set (reference) and a corresponding base image (input) using E_p^1, E_p^2, E_h with $\delta = 8$ and $D = 8$. We show registration results in Fig. 5.

7 Discussion

Experimental results show the proposed energy model generally outperforms the previous methods with pairwise spatial prior. In Table 1, d_{rms} indicates

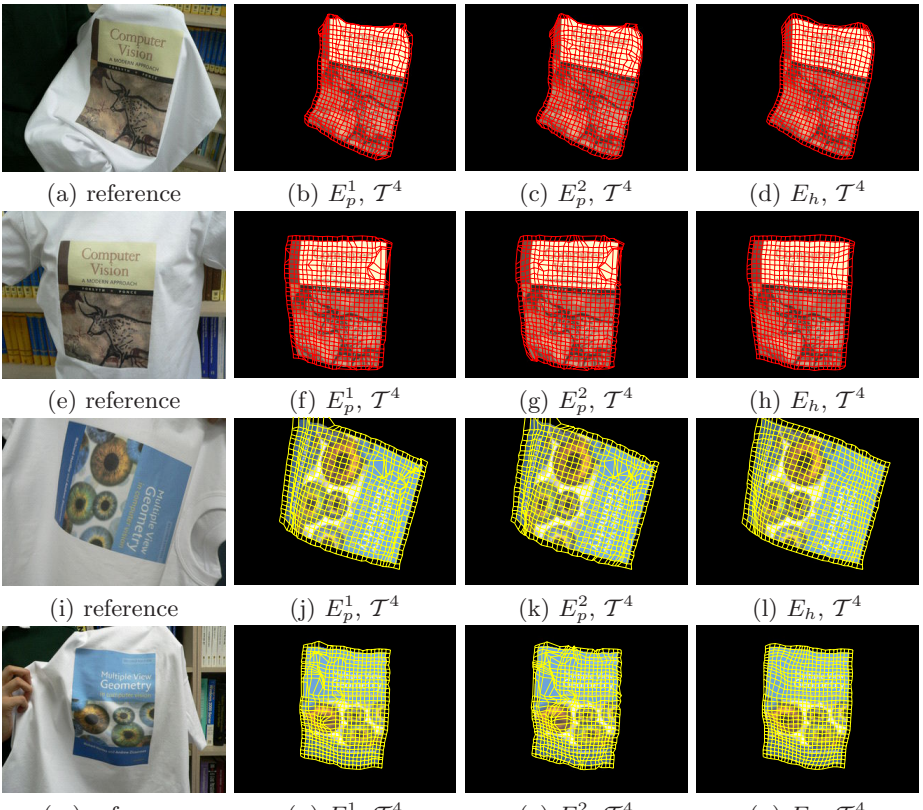


Fig. 5. Results from real data. {(b),(f),(j),(n)} and {(c),(g),(k),(o)} are generated by the pairwise models E_p^1 and E_p^2, respectively. {(d),(h),(l),(p)} are generated by the proposed model E_h.

the overall accuracy while d_{max} implies the reliability of the methods. Therefore the result on $\mathcal{T}^4$ shows the proposed model is more accurate and reliable than the pairwise models. Although d_{rms} in $\mathcal{T}^4$ are usually better than d_{rms} in $\mathcal{T}^2$ for every model, differences of d_{max} of pairwise models are marginal while the proposed model decreases d_{max} in successive registrations. From this fact, we conclude the proposed dynamic energy model increases the accuracy while assuring the reliability.

In addition to the quantitative analysis, the quality of the generated mesh using the proposed model is superior to the pairwise models. As can be seen in Fig. 4, the resulting meshes of pairwise models present irregular grids, however the meshes of the proposed model is closely recovering the ground truths. This situation is more distinctive in experiments with real data. Due to the illumination difference and the noise, the data cost is more erroneous, so the registration depends more on the smoothness cost. In Fig. 5, the resulting displacements of pairwise energy model are prone to converge to wrong places. However in

the proposed model, this kind of errors are eliminated because the higher-order spatial prior enforces the smoothness of the surface deformation.

References

1. Zitova, B., Flusser, J.: Image registration methods: a survey. Image and Vision Computing 21(11), 977–1000 (2003)
2. Rohr, K.: Image Registration Based on Thin-Plate Splines and Local Estimates of Anisotropic Landmark Localization Uncertainties. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) MICCAI 1998. LNCS, vol. 1496. Springer, Heidelberg (1998)
3. Pilet, J., Lepetit, V., Fua, P.: Real-Time Non-Rigid Surface Detection. In: CVPR (2005)
4. Kwon, D., Yun, I.D., Lee, K.H., Lee, S.U.: Efficient Feature-Based Nonrigid Registration of Multiphase Liver CT Volumes. In: BMVC (2008)
5. Rueckert, D., Sonoda, L.I., Hayes, C., Hill, D.L.G., Leach, M.O., Hawkes, D.J.: Nonrigid Registration Using Free-Form Deformations: Application to Breast MR Images. IEEE Trans. Medical Imaging 18(8), 712–721 (1999)
6. Glocker, B., Komodakis, N., Paragios, N., Tziritas, G., Navab, N.: Inter and Intramodal Deformable Registration: Continuous Deformations Meet Efficient Optimal Linear Programming. In: Karssemeijer, N., Lelieveldt, B. (eds.) IPMI 2007. LNCS, vol. 4584, pp. 408–420. Springer, Heidelberg (2007)
7. Shekhovtsov, A., Kovtun, I., Hlavác, V.: Efficient MRF Deformation Model for Non-Rigid Image Matching. In: CVPR (2007)
8. Kolmogorov, V.: Convergent Tree-Reweighted Message Passing for Energy Minimization. IEEE Trans. Pattern Anal. Mach. Intell. 28(10), 1568–1583 (2006)
9. Komodakis, N., Tziritas, G., Paragios, N.: Fast, Approximately Optimal Solutions for Single and Dynamic MRFs. In: CVPR (2007)
10. Sederberg, T.W., Parry, S.R.: Free-form deformation of solid geometric models. ACM SIGGRAPH Computer Graphics 20(4), 151–160 (1986)
11. Wainwright, M.J., Jaakkola, T., Willsky, A.S.: MAP estimation via agreement on trees: message-passing and linear programming. IEEE Trans. on Information Theory 51(11), 3697–3717 (2005)
12. Kschischang, F.R., Frey, B.J., Loeliger, H.A.: Factor graphs and the sum-product algorithm. IEEE Trans. on Information Theory 47(2), 498–519 (2001)
13. Bookstein, F.: Principal warps: Thin-plate splines and the decomposition of deformations. IEEE Trans. Pattern Anal. Machine Intell. 11(6), 567–585 (1989)
14. Kohli, P., Torr, P.H.: Efficiently Solving Dynamic Markov Random Fields using Graph Cuts. In: ICCV (2005)
15. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Constructing free-energy approximations and generalized belief propagation algorithms. IEEE Trans. on Information Theory 51(7), 2282–2312 (2005)
16. Potetz, B.: Efficient belief propagation for vision using linear constraint nodes. In: CVPR (2007)
17. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., Rother, C.: A Comparative Study of Energy Minimization Methods for Markov Random Fields. In: ECCV (2006)
18. Weiss, Y., Freeman, W.T.: On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. IEEE Trans. on Information Theory 47(2), 736–744 (2001)
19. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. Int. J. Computer Vision 60(2), 91–110 (2004)

Tracking of Abrupt Motion Using Wang-Landau Monte Carlo Estimation

Junseok Kwon and Kyoung Mu Lee

Department of EECS, ASRI, Seoul National University,151-742, Seoul, Korea
paradis0@snu.ac.kr, kyoungmu@snu.ac.kr

Abstract. We propose a novel tracking algorithm based on the Wang-Landau Monte Carlo sampling method which efficiently deals with the abrupt motions. Abrupt motions could cause conventional tracking methods to fail since they violate the motion smoothness constraint. To address this problem, we introduce the Wang-Landau algorithm that has been recently proposed in statistical physics, and integrate this algorithm into the Markov Chain Monte Carlo based tracking method. Our tracking method alleviates the motion smoothness constraint utilizing both the likelihood term and the density of states term, which is estimated by the Wang-Landau algorithm. The likelihood term helps to improve the accuracy in tracking smooth motions, while the density of states term captures abrupt motions robustly. Experimental results reveal that our approach efficiently samples the object's states even in a whole state space without loss of time. Therefore, it tracks the object of which motion is drastically changing, accurately and robustly.

1 Introduction

Object tracking is a well known problem to computer vision community [1]. Visual tracking has been utilized in surveillance systems and other intelligent vision systems. Recently, many of the visual tracking systems trends have been towards addressing complex outdoor videos rather than lab environmental ones. These complex outdoor videos which can be easily found in web sites, usually contain drastically abrupt motions.

Traditional tracking methods can be divided into two categories: the sampling based method (stochastic approach) and the detection based method (deterministic approach). In the stochastic approach, the particle filter (PF) has shown efficiency in handling non-gaussianity and multi-modality [2,3]. In multi-object tracking, Markov Chain Monte Carlo (MCMC) reduces computational costs to deal with high-dimensional state space [4,5]. Data-Driven MCMC provides quick convergence results with efficient proposals [6]. The stochastic approach has an advantage of reflecting the motion's uncertainty. In the deterministic approach, the Adaboost detector has been widely used in detecting a target object [7] and various data association techniques have been applied to connect the detected target and make a trajectory [8]. The deterministic approach usually provides reliable results by utilizing the bottom-up information. In general, both these

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 387–400, 2008.

(a) Frame #246 (b) Frame #247 (c) Frame #248 (d) Frame #249

Fig. 1. Example of an abrupt motion. The camera shot change causes the boxer to have an abrupt motion at consecutive frames, (b)-(c).

two tracking approaches basically assume that the appearance and motion of an object are smoothly changed over time.

However, in many complex outdoor scenarios, these motion and appearance smoothness constraints are frequently violated. Recently, online feature selection techniques have started to tackle this problem [9,10,11]. New features are selected online to adapt abrupt changes in appearance. Yet, most of tracking methods rarely consider abrupt motions which cause traditional algorithms to fail. In this study, we address the problem of tracking objects whose motion is mostly smooth, but which changes rapidly over one or more small temporal intervals. This motion typically occurs in two challenging situations: (1) video consists of edited clips acquired from several cameras (shot change), (2) object or camera rapidly moves. Figure 1 illustrates an example of the first situation.

The philosophy of our method is that two kinds of the motion, which is smooth or abrupt, can be efficiently tracked at the same time by trading off two factors which are the likelihood term and the density of states term. If the likelihood term is highly weighted, our method is similar with conventional tracking methods which track the smooth motion. On the other hand, if the density of states term is highly weighted, the method has the similar properties of detection methods which could capture the abrupt motion. So, as trading off these two terms, our method combines merits of tracking methods with ones of detection methods.

The first contribution of this paper is that, to the best of our knowledge, we firstly introduce the Wang-Landau Monte Carlo (WLMC) sampling method to the tracking problem. The WLMC sampling method was recently proposed in the statistical physics literature, which accurately estimates the density of states [12]. The second contribution is to propose the WLMC based tracking method and provide the unified framework to track both smooth and abrupt motion without loss of time. In the unified framework, the method utilizes the efficient sampling schedule. The schedule encourages to sample less-visited regions of the state space, while spending more time refining local maxima. And the method provides a statistical way to reach the global maximum. The third contribution is that, in order to design more efficient sampling scheme, we modify the WLMC sampling method into an annealed version and present the annealed WLMC based tracking method. The method searches for interesting subregions of the state space by employing the density of states and reduces the state space to these subregions where the target exists.

2 Related Works

The quasi-random sampling addresses the problem of tracking pedestrians from a moving car [13]. To cope with the abrupt changes in motion and shape, the method combines particle filter with quasi-random sampling. This algorithm has two drawbacks. First, the method chooses highly weighted particles and densely samples new states around the states of those particles. However, if there are a few deeper local maxima, most of samples get trapped in those local maxima. Second, the method uses uniform sampling over the entire state space to capture the abrupt changes. However, if entire state space is very large, uniform sampling scheme can be wasteful.

The cascade particle filter addresses tracking in low frame rate videos [14]. In this approach, the detection algorithm is well combined with particle filter to deal with abrupt motions. It demonstrates efficiency in a face tracking case. However, this approach requires complex observers and an offline training process. On the other hand, we consider the human as a target object, which makes it more challenging than the face, and treat much larger areas in an image for tracking.

3 Wang-Landau Monte Carlo Algorithm

The density of states is the number of states which belongs to a given energy subregion. Let us consider the 2D Ising model and assume that the energy function of the model is defined by only pairwise term [12]. Then, at the lowest energy, the density of states is 2 because the states which yield the lowest energy, are following 2 cases; all nodes have same values, -1 or +1. As it is intractable to accurately calculate the density of states in all energy subregions, the WLMC method [12] approximately estimate the density of states through a Monte Carlo simulation. Let us assume that the energy space $\mathbf{E}$ is divided into d disjoint subregions such that

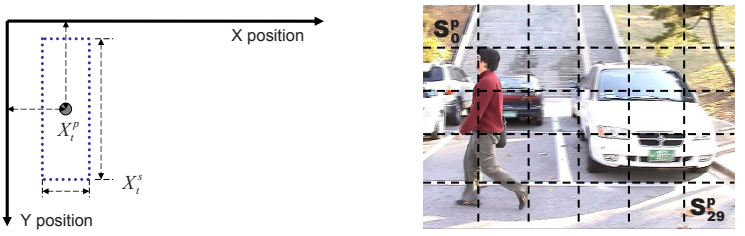
$$E_i \cap E_j = \phi \quad for \quad i \neq j,\ i, j \in \{1, \ldots, d\} \quad and \sum_{all\ i}^{d} E_i = \mathbf{E}. \tag{1}$$

Each energy subregion is visited via random walk in the energy space. If E_i is visited, we increase the histogram $h(E_i)$ by one and modify the density of states $g(E_i)$ by a modification factor $f > 1$. One proper modification method is

$$g(E_i) \leftarrow g(E_i) * f, \tag{2}$$

where $g(E_i)$ is initially set to 1 for all i and gradually updated by (2). While the simulation progresses, the random walk produces a flat histogram over the energy space. Note that flat has a different meaning; As described in [12], if the lowest bin of the histogram is larger than 80% of the histogram average, we consider the histogram is flat. Since the flat histogram means that all energy subregions are visited at least to some degree, the algorithm proceeds to the next random walk in a coarse-to-fine manner.

$$f \leftarrow \sqrt{f}. \tag{3}$$



(a) Example of (X_t^p, X_t^s) (b) Example of S_i^p

Fig. 2. Example of a state and subregion. (a) X_t^p represents the center of an object and X_t^s indicates the size of the boundary box. (b) $\mathbf{S}^p$ is divided into 30 equal-size subregions.

The modification factor is reduced to a finer version by (3), and the histogram is reset. Simulation continues until the histogram becomes flat again and restarts with a finer modification factor. The algorithm is terminated when the factor becomes highly close to 1 or the number of iterations reaches a predefined value.

In the WLMC simulation, a new state is proposed at each time. This new state is accepted or rejected according to the transition probability. The transition probability of the current state from E_i to E_j is defined by

$$p(E_i \rightarrow E_j) = \min\left[1, \frac{1/g(E_j)}{1/g(E_i)}\right]. \tag{4}$$

Note that the transition probability is calculated with the inverse of the density of states. This means that the transition is guided to less visited energy subregions.

4 WLMC Based Tracking Method

4.1 Preliminary

The state $\mathbf{X}_t$ at time t consists of the position and scale of an object; $\mathbf{X}_t = (X_t^p, X_t^s)$. And the state space $\mathbf{S}$ is defined by a set of all possible states. This state space $\mathbf{S}$ can be decomposed into the state space of position and scale; $\mathbf{S} = \mathbf{S}^p \times \mathbf{S}^s$. As, in many cases, an abrupt motion occurs by the change of the position, we assume that the scale of an object is smooth over time and only consider abrupt changes in $\mathbf{S}^p$. $\mathbf{S}^p$ is then divided into d disjoint subregions; $S_i^p, i = \{1, \ldots, d\}$. A simple dividing strategy is to partition $\mathbf{S}^p$ into equal-size grids as shown in Figure 2(b). Note that, to adapt the WLMC method to an image-based tracking problem, we replace E_i in Section 3 with S_i^p and calculate the density of states at each S_i^p. Our method can be easily extended to deal with abrupt changes of the scale by calculating the density of states at $\mathbf{S}^s$.

4.2 Bayesian Object Tracking Approach

The object tracking problem is usually formulated as the Bayesian filtering. Given the state of an object at time t, $\mathbf{X}_t$ and the observation up to time t,

$\mathbf{Y}_{1:t}$, the Bayesian filter updates the posteriori probability $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ with the following rule:

$$p(\mathbf{X}_t|\mathbf{Y}_{1:t}) = cp(\mathbf{Y}_t|\mathbf{X}_t) \int p(\mathbf{X}_t|\mathbf{X}_{t-1})p(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1})d\mathbf{X}_{t-1}, \tag{5}$$

where $p(\mathbf{Y}_t|\mathbf{X}_t)$ is the observation model that measures the similarity between the observation at the estimated state and the given model; $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ is the transition model which predicts the next state $\mathbf{X}_t$ based on the previous state $\mathbf{X}_{t-1}$, and; c is the normalization constant. The observation model generally utilizes color, edges or texture as a feature [1]. With the posteriori probability $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ computed by the observation model and the transition model, we obtain the Maximum a Posteriori (MAP) estimate over the N number of samples at each time t.

$$\mathbf{X}_t^{MAP} = \underset{\mathbf{X}_t^n}{\arg\max}\, p(\mathbf{X}_t^n|\mathbf{Y}_{1:t}) \quad for\ n = 1, \ldots, N, \tag{6}$$

where $\mathbf{X}_t^{MAP}$ denotes the best configuration which can explain the current state with the given observation. However note that the integration in (5) is unfeasible in high dimensional state space. To address this problem, we use the Metropolis Hastings (MH) algorithm that is one of the popular MCMC method. The MH algorithm consists of two main steps; proposal step and acceptance step.

In this work, the traditional transition model $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ is reinforced by the approximated prior term $p^*(\mathbf{X}_t)$ to track the abrupt motion. Then our transition model is defined by

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) \approx p(\mathbf{X}_t|\mathbf{X}_{t-1})\frac{p^*(\mathbf{X}_t)}{p(\mathbf{X}_t)} = cp(\mathbf{X}_t|\mathbf{X}_{t-1})p^*(\mathbf{X}_t), \tag{7}$$

where the inverse of the prior term $p(\mathbf{X}_t)$ is replaced with constant c. We sequentially estimate the approximated prior term $p^*(\mathbf{X}_t)$ using the density of states that is calculated by the Wang-Landau recursion step.

4.3 Proposal Step

The proposal density designs the transition from a given state to a new state based on some prior knowledge about the motion. Our prior knowledge on a motion is that objects can go anywhere in a scene even at one proposal step. With this assumption, we design a new proposal density that covers the whole state space and proposes highly diverse states.

$$Q(\mathbf{X}_t'; \mathbf{X}_t) = \begin{cases} Q_{AR}(X_t^{s'}; X_t^s) & \text{for the scale} \\ Q_U(X_t^{p'}) & \text{for the position} \end{cases}. \tag{8}$$

Q_{AR} proposes a new scale state $X_t^{s'}$ based on X_t^s with a second-order autoregressive process. This process well describes the characteristic of the smooth

change in scale [2], and fits our smoothness assumption of the scale. To propose a new position state, Q_U utilizes two steps. The first step randomly chooses one subregion S_i^p to obtain diverse states which cover abrupt changes in position. And the second step uniformly proposes a new state $X_t^{p'}$ over the chosen S_i^p to simulate the density of states at S_i^p.

For the success of our proposal step, its efficiency has to be considered. The proposal density (8) can be wasteful if it proposes numerous inefficient states where the probabilities that the target exists are very low. Hence, our algorithm addresses this inefficiency utilizing the density of states in the acceptance step.

4.4 Acceptance Step

The acceptance step determines whether the proposed state is accepted or not and can be simply calculated by the likelihood ratio between the current and proposed states as follows.

$$a = min\left[1, \frac{p(\mathbf{Y}_t|\mathbf{X}'_t)Q(\mathbf{X}_t;\mathbf{X}'_t)}{p(\mathbf{Y}_t|\mathbf{X}_t)Q(\mathbf{X}'_t;\mathbf{X}_t)}\right], \tag{9}$$

where $p(\mathbf{Y}_t|\mathbf{X}'_t)$ denotes the likelihood term over the state $\mathbf{X}'_t$ and $Q(\mathbf{X}'_t;\mathbf{X}_t)$ represents the proposal density.

Our algorithm combines the density of states term with the acceptance ratio in (9). Let M be a mapping function from the state $\mathbf{X}_t$ to the subregion S_i^p which contains the position state, $\mathbf{X}_t^p$ of $\mathbf{X}_t$.

$$M : \mathbf{X}_t \rightarrow \mathbf{S}_i^p. \tag{10}$$

Then the modified acceptance ratio is

$$a = min\left[1, \frac{p(\mathbf{Y}_t|\mathbf{X}'_t)^\alpha p^*(\mathbf{X}'_t)Q(\mathbf{X}_t;\mathbf{X}'_t)}{p(\mathbf{Y}_t|\mathbf{X}_t)^\alpha p^*(\mathbf{X}_t)Q(\mathbf{X}'_t;\mathbf{X}_t)}\right] = min\left[1, \frac{\frac{p(\mathbf{Y}_t|\mathbf{X}'_t)^\alpha}{g(M(\mathbf{X}'_t))}Q(\mathbf{X}_t;\mathbf{X}'_t)}{\frac{p(\mathbf{Y}_t|\mathbf{X}_t)^\alpha}{g(M(\mathbf{X}_t))}Q(\mathbf{X}'_t;\mathbf{X}_t)}\right], \tag{11}$$

where $p^*(\mathbf{X}'_t)$ denotes the approximated prior term in (7), $g(M(\mathbf{X}'_t))$ expresses the density of states at the subregion that includes the position state $\mathbf{X}_t^{p'}$ of $\mathbf{X}'_t$, and α indicates the weighting parameter. Our acceptance ratio in (11) has two different properties compared to that in (9). The first property is that (11) provides a way to escape from local maxima and reach to the global maximum. This property is crucial to the success of our tracking algorithm. If an abrupt motion exists in a scene, the algorithm has to sample the states in larger areas to deal with that motion where the Markov Chain has higher chances of meeting local maxima. In our acceptance step, the Markov Chain is guided by the ratio between the likelihood score and the density of states score, $\frac{p(\mathbf{Y}_t|\mathbf{X}_t)}{g(M(\mathbf{X}_t))}$. At a local maximum, this ratio initially has a large value since the likelihood has the higher score around the local maximum. However, while the simulation goes on, the ratio continues to decrease as the density of states generally increases around

(a) Escaping process (b) Tracking results (c) Density of states

Fig. 3. Properties of our acceptance ratio. (a) If the density of states in region 3 is much larger than one in region 4, the proposed state can be accepted although the state has a lower likelihood score than that of current state. (c) The brighter the color, the larger the density of states. Our method gets more samples at regions of two boxers while exploring all subregions at least to some degree.

local maxima. The proposed state is accepted when the ratio over the current state sufficiently decreases compared to one over the proposed state. Figure 3(a) illustrates the process of escaping from the local maximum.

As the second property, (11) efficiently schedules a sampling procedure so that the Markov Chain resides in a local maximum for a longer period, while guaranteeing to explore the whole state space at least to some degree. This property provides increased flexibility over the proposal density in (8). Note that the density of states term allows chances for the proposed state to be accepted at rarely visited subregions. On the other hand, the likelihood term forces the proposed state to be frequently accepted around local maxima. Since the length of the Markov Chain is limited, these two terms form the trade-off relationship. α in (11) controls this trade-off relationship. Higher weights on the likelihood term result in increasing the accuracy of MAP estimate. Conversely, the density of states term has to be increasingly weighted to cover the whole state space. Our acceptance ratio efficiently deals with this trade-off relationship as shown in Figure 3(c).

4.5 Wang-Landau Recursion Step

In this section, we propose a new efficient step called the Wang-Landau recursion to calculate the density of states $g(M(\mathbf{X}_t))$ in (11). This step follows the similar procedure as in the original Wang -Landau algorithm discussed in Section 3. Figure 4 provides the detailed process of our WLMC based tracking method that include the proposal, acceptance and Wang-Landau recursion step. This figure shows how the density of states is adapted for our tracking problem.

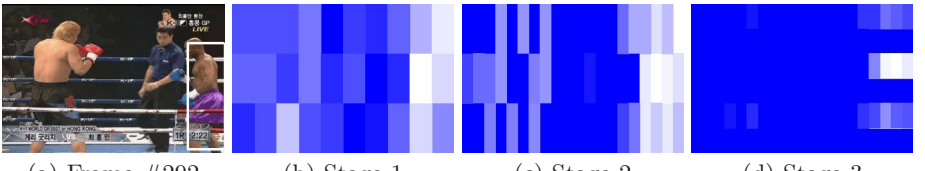
The key point is that the Wang-Landau recursion step addresses the chicken-egg-type problem. In order to estimate the density of states accurately, the acceptance ratio in (11) should guide the Markov Chain in the direction of producing a flat histogram over the whole state space. While, to calculate the acceptance ratio in (11), the density of states has to be known in advance. This recursion step provides a systematic way to visit all the subregions at least to some degree and

1) Initialize the Wang-Landau recursion:
Given d disjoint subregions S_i^p,
$g(S_i^p)=1,\ h(S_i^p)=0 \quad for\ i=1,\ldots,d$ and set $f=f_0=2.7$.

2) MCMC sampling: Repeat N times, where N is the total number of samples
a) Given the current state X_t^n (n-th sample at time t),
propose the new state X_t' using proposal density (8).
b) Compute the acceptance ratio (11),
If accepted, $X_t^{n+1}=X_t'$, other wise, $X_t^{n+1}=X_t^n$.
c) Wang-Landau update:
Update $g(M(X_t^{n+1}))\leftarrow g(M(X_t^{n+1}))*f \ ,h(M(X_t^{n+1}))\leftarrow h(M(X_t^{n+1}))+1$.
If the histogram is flat, reinitialize $h(S_i^p)=0$ for all i and set $f\leftarrow\sqrt{f}$.

3) Compute the MAP estimate X_t^{MAP}.

Fig. 4. WLMC based tracking method



(a) Frame #292 (b) Stage 1 (c) Stage 2 (d) Stage 3

Fig. 5. Annealing process. A-WLMC sequentially reduces $\mathbf{S}^p$ from (b) to (d) using the density of states. Then, A-WLMC leaves some subregions that contain robust candidates of the target position and eventually tracks the target as shown in (a).

simultaneously acquire the exact density of states. Note that the Wang-Landau algorithm typically converges although it does not satisfy detailed balance [15].

5 Annealed WLMC Based Tracking Method

We extend our WLMC based tracking method to an annealed version (A-WLMC) for more efficient sampling. In A-WLMC, the Markov Chain is defined over the annealed state space. And A-WLMC concentrates sampling on theses annealed subregions that compactly contain the target object. Figure 5 shows the process on how the state space is reduced to the interesting subregions. The algorithm starts the process over the whole state space and performs each stage using the WLMC based tracking method presented in Section 4. At the end of each stage, the state space is reduced by half, and the Chain is restarted over the reduced state space.

We utilize the density of states to anneal the state space. The state space basically consists of d disjoint subregions. Since the density of states becomes larger

1) Process the WLMC based tracking method.

2) Annealing step: if the histogram is flat,

a) Choose $d/2$ number of subregions S_i^p according to $g(S_i^p)$ in descending order and represent the chosen subregions as S_i^c *for* $i = 1, \ldots d/2$.

b) Divide each S_i^c into two regions: $S_{i_1}^c$ and $S_{i_2}^c$.

c) Update the density of states and subregions.

$$S_{2i-1}^a = S_{i_1}^c, \; S_{2i}^a = S_{i_2}^c \quad \textit{for} \quad i = 1, \ldots d/2.$$

$$g(S_{2i-1}^a) = g(S_{2i}^a) = g(S_i^c) \quad \textit{for} \quad i = 1, \ldots d/2.$$

$$S_i^p = S_i^a \quad \textit{for} \quad i = 1, \ldots d.$$

where S_i^a represents the annealed subregion.

Fig. 6. Annealed WLMC based tracking method

around the local maxima, we choose the $d/2$ number of subregions according to the density of states in descending order. The chosen subregions are individually divided into two regions so that the total number of subregions becomes d again. The overall procedure of the annealed version is summarized in Figure 6.

6 Experimental Result

In this paper, the observation model utilized the HSV color histogram as a feature which is known to be robust to the illumination changes, and Bhattacharyya coefficient as a similarity measure [16]. We tested three video sequences: Seq.1, Seq.2 for camera shot changes, and; Seq.3 for rapid motions.[1] For the fair comparison, we used equal number of samples; 600, and compared the proposed algorithm with five different tracking methods: standard MCMC is based on [4]. Proposal variances are separately set to 8, 4 for the x, y direction; Adaptive MCMC is based on [17]; Quasi-random sampling is based on [13]; Particle filter is based on [3]. The motion model utilized the second-order autoregressive process and noise model is defined by the gaussian function of which the variance is set to 250; Mean shift is based on the implemented function in opencv.

6.1 Quantitative Comparison

Coverage test: The recall ρ and the precision ν measure the configuration errors between the ground truth state and the estimated state.

$$\rho = \frac{A_t^X \cap A_t^G}{A_t^G}, \quad \nu = \frac{A_t^X \cap A_t^G}{A_t^X}, \tag{12}$$

[1] The tracking results are available at $http://cv.snu.ac.kr/WLMC\_tracking.html$.

Table 1. ***F-measure*** in Seq. 1. As α value increases, the weight on the likelihood term in (11) also increases.

α	A-WLMC	Adaptive MCMC	Standard MCMC	Quasi-random	Particle filter
0.5	0.783126	0.780664	0.773494	0.726511	0.715354
1.0	0.795748	0.774263	0.769159	0.726511	0.715354
1.5	0.816221	0.773265	0.756279	0.726511	0.715354

where A_t^X denotes the estimated area and A_t^G indicates the ground truth area at time t. For good tracking quality, both the recall and precision should have high values. In information retrieval literatures, *F-measure* is often used for evaluating this quantity.

$$F = \frac{2\nu\rho}{\nu + \rho}. \tag{13}$$

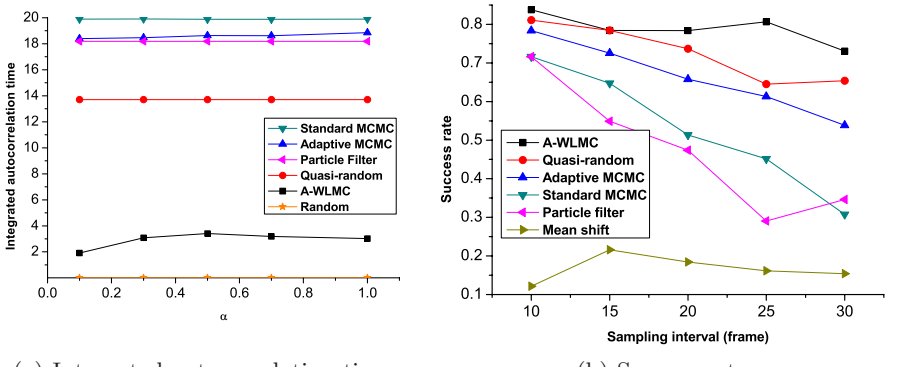
When the ground and estimated area perfectly overlap, *F-measure* is 1.0.

We obtained the ground truth states by manually drawing the bounding box around the target. Note that, in other methods, the state is re-initialized to the ground truth before they fail to track the abrupt motion. Then, the results in Table 1 indicates the accuracy of tracking the *smooth* motion, and states that the A-WLMC method is also as good as existing methods in the smooth motion case.

Autocorrelation time: The autocorrelation time measures the degree of statistical independence between samples. This independence property is important in terms of sampling efficiency. If samples are highly correlated, the statistical error does not decrease at the rate of the square root of the number of samples.

Let us define the autocorrelation function as follows:

$$C_{xx}(k) = \mathbb{E}\left[(\mathbf{X}_t^n - \mathbb{E}[\mathbf{X}_t^n])(\mathbf{X}_t^{n+k} - \mathbb{E}[\mathbf{X}_t^{n+k}])\right], \tag{14}$$



(a) Integrated autocorrelation time (b) Success rate

Fig. 7. Evaluation of the tracking methods. (a) Integrated autocorrelation time at Seq. 2. (b) Success rate at Seq. 3 as a function of down-sampling interval for different tracking methods.

where $\mathbb{E}$ is the expectation operator, $\mathbf{X}_t^n$ and $\mathbf{X}_t^{n+k}$ are the n-th and $(n+k)$-th samples at time t , respectively. This function generally decays exponentially by the number of samples k such that,

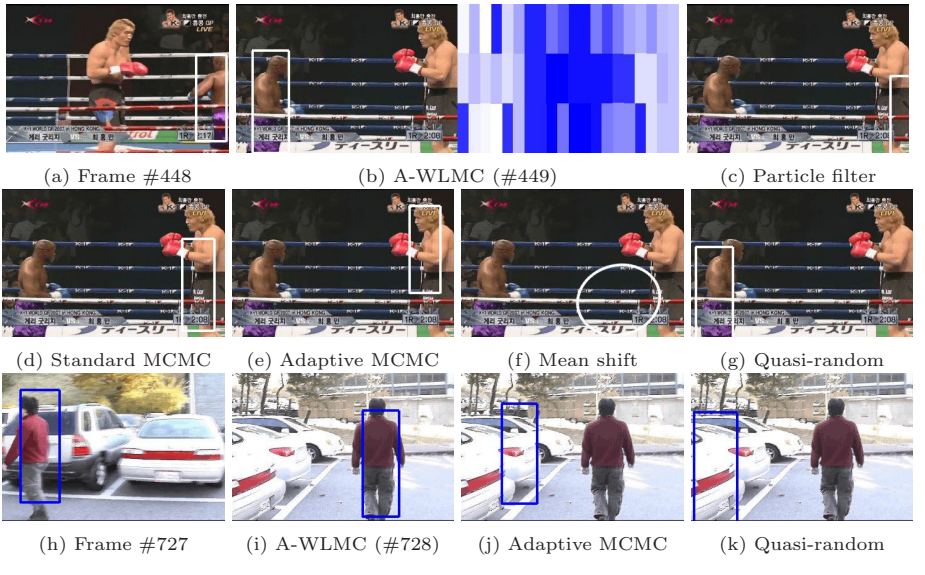
$$C_{xx}(k) \approx \exp(-\frac{k}{\tau_{exp}}), \tag{15}$$

where τ_{exp} is the *exponential autocorrelation time*. For the computational simplicity, we use *integrated autocorrelation time* τ_{int} suggested by [18].

$$\int_0^\infty C_{xx}(k)dt = \int_0^\infty C_{xx}(0)\exp(-\frac{k}{\tau_{int}})dk = \tau_{int}C_{xx}(0), \tau_{int} = \sum_k \frac{C_{xx}(k)}{C_{xx}(0)}. \tag{16}$$

Figure 7(a) displays the efficiency of the A-WLMC method to produce both uncorrelated and meaningful samples. Although the random sampling method is the best in terms of statistical independence, this method is not guided in a principled manner. In contrast, A-WLMC is the winner in terms of efficiency. A-WLMC guarantees samples to converge to the target density and simultaneously generates higher uncorrelated samples than those by other methods.

Success rate: If *F-measure* is larger than 0.5, the target is considered as correctly tracked at that frame. The success rate indicates the ratio between the number of correctly tracked frames and the number of total frames. For this test we down-sampled Seq. 3 with the sampling interval from 10 frames to 30 frames. The results are depicted in Figure 7(b). In comparison with the other results, the success rate of the A-WLMC method is less affected by the change of the sampling interval, whereas those of other methods rapidly decrease as the sampling

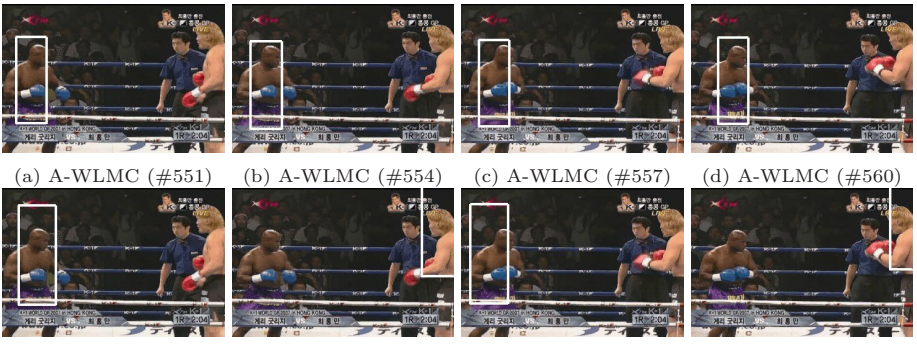


(a) Frame #448 (b) A-WLMC (#449) (c) Particle filter

(d) Standard MCMC (e) Adaptive MCMC (f) Mean shift (g) Quasi-random

(h) Frame #727 (i) A-WLMC (#728) (j) Adaptive MCMC (k) Quasi-random

Fig. 8. Tracking results when the camera shot change occurs in Seq. 2, Seq. 1



(a) A-WLMC (#551) (b) A-WLMC (#554) (c) A-WLMC (#557) (d) A-WLMC (#560)

(e) Quasi-rand (#551) (f) Quasi-rand (#554) (g) Quasi-rand (#557) (h) Quasi-rand (#560)

Fig. 9. Tracking results of the smooth motion case in Seq. 2

(a) A-WLMC (#18) (b) A-WLMC (#22) (c) A-WLMC (#25)

Fig. 10. Tracking results in videos where rapid motions exist. In Seq. 3, the video is down-sampled to 25 sampling interval.

interval increases. It is significant to note that A-WLMC successfully tracks the target even in highly down-sampled video which contains severe abrupt motions.

Speed: A-WLMC has no additive computational burden compared to the other sampling based tracking methods since the density of states can be calculated at extremely less computational cost. A-WLMC runs at 1∼10 fps for 320×240 videos. Note that our code is not optimized.

6.2 Qualitative Comparison

Figure 8 presents the tracking results in the camera shot change case. In the video, A-WLMC successfully tracked the target whereas other methods failed to escape from the previous position of the target. The quasi-random method also tracked the abrupt motion in Seq. 2 (Figure 8(g)) whereas the method failed to track the motion in Seq. 1 (Figure 8(k)). We also illustrate the density of states obtained at frame #449 of Seq. 2 in the right part of Figure 8(b). The whiter regions indicate that A-WLMC got more samples at those regions which can be regarded as local maxima. As shown in the figure, there are a number of local minima found by A-WLMC. This means that our method has a ability to escape one local maximum and reach to another one. Furthermore, we

tested to track the target of which motion is smooth. As A-WLMC and quasi-random sampled states at the larger portions of the state space compared with the conventional tracking methods, it is very crucial to check the accuracy of tracking the smooth motion and robustness to the clutters. As shown in Figure 9, A-WLMC accurately tracked the target of which motion is smooth over time. In contrast, the quasi-random sampling was easily distracted by clutter and often got trapped in local maxima which are the right boxer at the video. Figure 8 and 9 demonstrate that A-WLMC well tracks the smooth and abrupt motion at the same time compared with the other tracking methods. Note that most of the tracking performance comes from the A-WLMC *filter* rather than randomness of the proposal density. Quasi-random also used the similar proposal density, but which results were worse. As another example of an abrupt motion, we tested down-sampled video that included rapid motions of which directions and distances were quite unpredictable. A-WLMC addressed this uncertainty of the motion and accurately proposed the object's position as shown in Figure 10.

7 Conclusion

In this paper, we have proposed an effective tracking algorithm based on the Wang-Landau Monte Carlo. The algorithm efficiently addresses tracking of abrupt motions while smooth motions are also accurately tracked. Experimental results demonstrated that the proposed algorithm outperformed conventional tracking algorithms in severe tracking environments. Our current algorithm has not fully considered the abrupt changes in appearance. We leave this problem to be addressed in future research studies.

Acknowledgement

This research was supported in part by the Defense Acquisition Program Administration and Agency for Defense Development, Korea, through the Image Information Research Center under the contract UD070007AD, and in part by the MKE (Ministry of Knowledge Economy), Korea under the ITRC (Information Technolgy Research Center) Support program supervised by the IITA (Institute of Information Technology Advancement) (IITA-2008-C1090-0801-0018)

References

1. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Comput. Surv. 38(4) (2006)
2. Cai, Y., de Freitas, N., Little, J.: Robust visual tracking for multiple targets. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 107–118. Springer, Heidelberg (2006)
3. Isard, M., Blake, A.: Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407. Springer, Heidelberg (1998)

4. Khan, Z., Balch, T., Dellaert, F.: An mcmc-based particle filter for tracking multiple interacting targets. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 279–290. Springer, Heidelberg (2004)
5. Smith, K., Perez, D.G., Odobez, J.: Using particles to track varying numbers of interacting people. In: CVPR (2005)
6. Zhao, T., Nevatia, R.: Tracking multiple humans in crowded environment. In: CVPR (2004)
7. Wu, B., Nevatia, R.: Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. IJCV 75(2), 247–266 (2007)
8. Nillius, P., Sullivan, J., Carlsson, S.: Multi-target tracking: Linking identities using bayesian network inference. In: CVPR (2006)
9. Jepson, A., Fleet, D., Maraghi, T.E.: Robust online appearance models for visual tracking. PAMI 25(10), 1296–1311 (2003)
10. Yang, M., Wu, Y.: Tracking non-stationary appearances and dynamic feature selection. In: CVPR (2005)
11. Han, B., Davis, L.: On-line density-based appearance modeling for object tracking. In: ICCV (2005)
12. Wang, F., Landau, D.: Efficient, multiple-range random walk algorithm to calculate the density of states. Phys. Rev. Lett. 86(10), 2050–2053 (2001)
13. Philomin, V., Duraiswami, R., Davis, L.: Quasi-Random Sampling for Condensation. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 134–149. Springer, Heidelberg (2000)
14. Li, Y., Ai, H., Yamashita, T., Lao, S., Kawade, M.: Tracking in low frame rate video: A cascade particle filter with discriminative observers of different lifespans. In: CVPR (2007)
15. Atchade, Y., Liu, J.: The wang-landau algorithm for monte carlo computation in general state spaces. Technical Report (2004)
16. Perez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In: Tistarelli, M., Bigun, J., Jain, A.K. (eds.) ECCV 2002. LNCS, vol. 2359, pp. 661–675. Springer, Heidelberg (2002)
17. Roberts, G., Rosenthal, J.: Examples of adaptive mcmc (preprint, 2006)
18. Berg, B.: Introduction to markov chain monte carlo simulations and their statistical analysis. Phys.Stat. Mech. (2004)

Surface Visibility Probabilities in 3D Cluttered Scenes

Michael S. Langer

School of Computer Science, McGill University
Montreal, H3A2A7, Canada
langer@cs.mcgill.ca
http://www.cim.mcgill.ca/~langer

Abstract. Many methods for 3D reconstruction in computer vision rely on probability models, for example, Bayesian reasoning. Here we introduce a probability model of surface visibilities in densely cluttered 3D scenes. The scenes consist of a large number of small surfaces distributed randomly in a 3D view volume. An example is the leaves or branches on a tree. We derive probabilities for surface visibility, instantaneous image velocity under egomotion, and binocular half–occlusions in these scenes. The probabilities depend on parameters such as scene depth, object size, 3D density, observer speed, and binocular baseline. We verify the correctness of our models using computer graphics simulations, and briefly discuss applications of the model to stereo and motion.

1 Introduction

The term *clutter scene* typically refers to a scene that contains many visible objects [2,18,6] distributed randomly over space. In this paper, we consider a particular type of cluttered scene, consisting of a large number of small surfaces distributed over a 3D volume. An example is the leaves or branches of a tree.

Reconstructing 3D geometry of a cluttered scene is a very challenging task because so many depth discontinuities and layers are present. Although some computer vision methods allow for many depth discontinuities, such methods typically assume only a small number of layers are present, typically two [28].

The goal of this paper is to better understand the *probabilistic* constraints of surface visibilities in such scenes. We study how visibility probabilities of surfaces depend on various geometric parameters, namely the area, depth and density of surfaces. We also examine how visibility probabilities depend on observer viewpoint. Such probability models are fundamental in many methods for computing optical flow [24,29,19] and stereo[1,3]. The hope is that the probability models could be used to improve these methods, for example, in a Bayesian framework by providing a *prior* on surface visibilities in 3D cluttered scenes.

2 Related Work

The probability models that we present are related to several visibility models that have appeared in the literature. We begin with a brief review.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 401–412, 2008.

The first models are those that describe atmospheric effects such as fog, clouds, and haze [15] and underwater [22]. In these models, the size of each occluder is so small and the number of occluders is so large that each occluder projects to an image area much smaller than a pixel. This is the domain of partial occlusion *i.e.* transparency [26]. This scenario may be thought of as a limiting case of what is considered in this paper. Another interesting case for which partial occlusion effects dominate the analysis is that of rain [5].

A second related model is the *dead leaves* model [11,23] which consists of a sequence of 2D objects such as squares or disks that are dropped into a 2D plane, creating occlusion relationships. Here one is typically concerned with the distribution of object boundaries and sizes [21,20,10,17]. The key difference between the dead leaves model and the one we present is that the dead leaves model assumes 2D shapes that lie in the same plane *i.e.* dead leaves on the ground, and thus assumes orthographic projection rather than perspective projection. In the dead leaves model, there is no notion that shapes lying at greater depths appear smaller in the image, as is the case considered in this paper of "living leaves" in 3D space and viewed under perspective projection. While the dead leaves model is sufficient for understanding certain aspects of visibility in 3D cluttered scenes, for example, probability of surface visibility along a single line of sight as a function of scene depth (see Sec. 3), it cannot describe other aspects such as how visibility changes when the observer moves (Sec. 4 and 5).

A third set of related models consider both occlusion and perspective, and combine geometric and statistical arguments (see two monographs [7,30]). Such models have been used in computer graphics in the context of hidden surface removal [14] as well as for rendering foliage under ambient occlusion lighting [8]. Similar geometric and statistical reasoning has been used in computer vision for planning the placement of cameras to maximize scene visibility in the presence of random object occlusions[12].

3 Static Monocular Observer

The model developed here is based on several assumptions. First, the scene consists of surfaces that are randomly distributed over a 3D view volume. Each surface is represented by a particular point (e.g. its center of mass). The distribution of these points assumed to be Poisson [4] with density η, which is the average number of surface centers per unit volume. We assume that the density η is constant over a 3D volume. We also ignore interactions between surface locations such as clustering effects that may be found in real cluttered scenes and the fact that real surfaces such as leaves cannot intersect.

According to the Poisson model, the probability that any volume V contains k surface centers is

$$p(k\ surface\ centers\ in\ V) = e^{-\eta V}\frac{(\eta V)^k}{k!}$$

and so the probability that there are no surface centers in the volume V is

$$p(no\ surface\ centers\ in\ V) = e^{-\eta V}. \tag{1}$$

This last expression will be used heavily in this paper. Let's consider two instantiations of this model.

3.1 Planar Patches

Suppose the surfaces are planar patches of area A such as in the first row of Figure 1. What can we say about visibility in this case? The expected total area of the surfaces per unit volume is ηA. If the surface normals of all patches were parallel to the line of sight, then the expected number of surfaces intersected by a unit line that is parallel to these surface normals would be ηA. (To see this, imagine cutting a surface of area ηA into pieces of unit area and stacking them perpendicular to the line.) The more general and interesting case is that the surface normals are oriented in a uniformly random direction. In this case, it is easy to show that the average number of surface intersections per unit length line is reduced by a factor $\frac{1}{2}$, so the expected number of surfaces intersected by a line of unit length is:

$$\lambda = \frac{\eta A}{2}.$$

Using standard arguments about Poisson distributions[4], one can show that the probability $p(z)dz$ that the first visible surface is in depth interval $[z, z+dz]$ is:

$$p(z)dz = e^{-\lambda z}\ \lambda dz.$$

The probability density $p(z) = \lambda e^{-\lambda z}$ is defined over $z \in [0, \infty]$. Often, though, we are interested in scenes whose objects are distributed over some finite volume beyond some distance z_0 from the viewer. In this case, the probability density is

$$p(z) = \lambda e^{-\lambda(z-z_0)} \quad . \tag{2}$$

In computer graphics terminology, z_0 is the distance to the near clipping plane.

3.2 Spheres

The mathematical model presented in Sec. 5 for binocular visibility will be easier to derive in the case that the objects are spheres of radius R, so we consider this case next. Consider an imaginary line segment with one endpoint at the viewer and the other endpoint at distance z. In order for the 3D scene point at this distance z to be visible from the viewer, no sphere center can fall in a cylinder[1] whose radius is the same R as above, and whose axis is the line segment of length z. Such a cylinder has volume

$$V = \pi R^2 z.$$

[1] To be more precise, we would need to consider a capped cylinder [30], namely a cylinder capped at both ends by two half spheres. However, the extra volume of the capped cylinder has only a small effect on the model, and so we ignore this detail.

As with the planar patch scene, if all surfaces lie beyond a distance z_0, then the cylinder that cannot contain any sphere centers has volume:

$$V = \pi R^2 (z - z_0). \tag{3}$$

In order for the first surface along the line of sight to be in the depth interval $[z, z + dz]$, two conditions must hold: first, the cylinder must contain no sphere centers, and second, a sphere center must fall in a small volume slice $\pi R^2 dz$ capping the cylinder. From the above expression for V and from Eq. (1), the probability that the first surface visible along the line of sight is in depth interval $[z, z + dz]$ is

$$p(z)dz = \eta\pi R^2 e^{-\eta\pi R^2 (z - z_0)}\, dz \tag{4}$$

This model is the same as Eq. (2) where now for the case of spheres we have

$$\lambda = \eta\pi R^2.$$

We next present computer graphics simulations to illustrate this model.

3.3 Experiments

Scenes were rendered with the computer graphics package RADIANCE[2] [9]. Each rendered scene consisted of a set of identical surfaces randomly distributed over the rectanguloid

$$(x, y, z) \in [-4, 4] \times [-4, 4] \times [2, 8]$$

i.e. near and far clipping planes at $z = 2, 8$, respectively. The width of the field of view was 30 degrees. Each rendered image was 512×512 pixels.

Results are presented for scenes consisting of either squares (planar surfaces) or spheres. For both types of scene, three scene densities were used, namely $\eta =$ 14, 56, 225 surfaces per unit volume. The object sizes for each of these densities were chosen to be large, medium and small, such that the λ value was the same over all scenes and so the theoretical $p(z)$ curves are the same as well. Specifically, for the sphere scenes, the radii were arbitrarily chosen to be $R = 0.1,\ .05,\ .025$. The corresponding areas for the square scenes were $2\pi R^2$ which kept the λ values the same for all scenes.

Figure 1 compares the model of $p(z)$ in Eqs. 2 and 4 with the average histogram of *visible depths* from the RADIANCE Z buffer over ten scenes. For each histogram, 15 bins were used to cover the depth range $z \in [2, 8]$. Standard errors for each histogram bin are also shown.

For the scenes with larger values of R, the standard errors within each histogram are greater than for the scenes with small R values. To understand this effect, consider for example the histogram bin for the nearest depth. The number of pixels in that histogram bin depends on the number of sphere centers that

[2] Unlike OpenGL, RADIANCE computes cast shadows which we will use later in the paper to examine binocular half-occlusions.

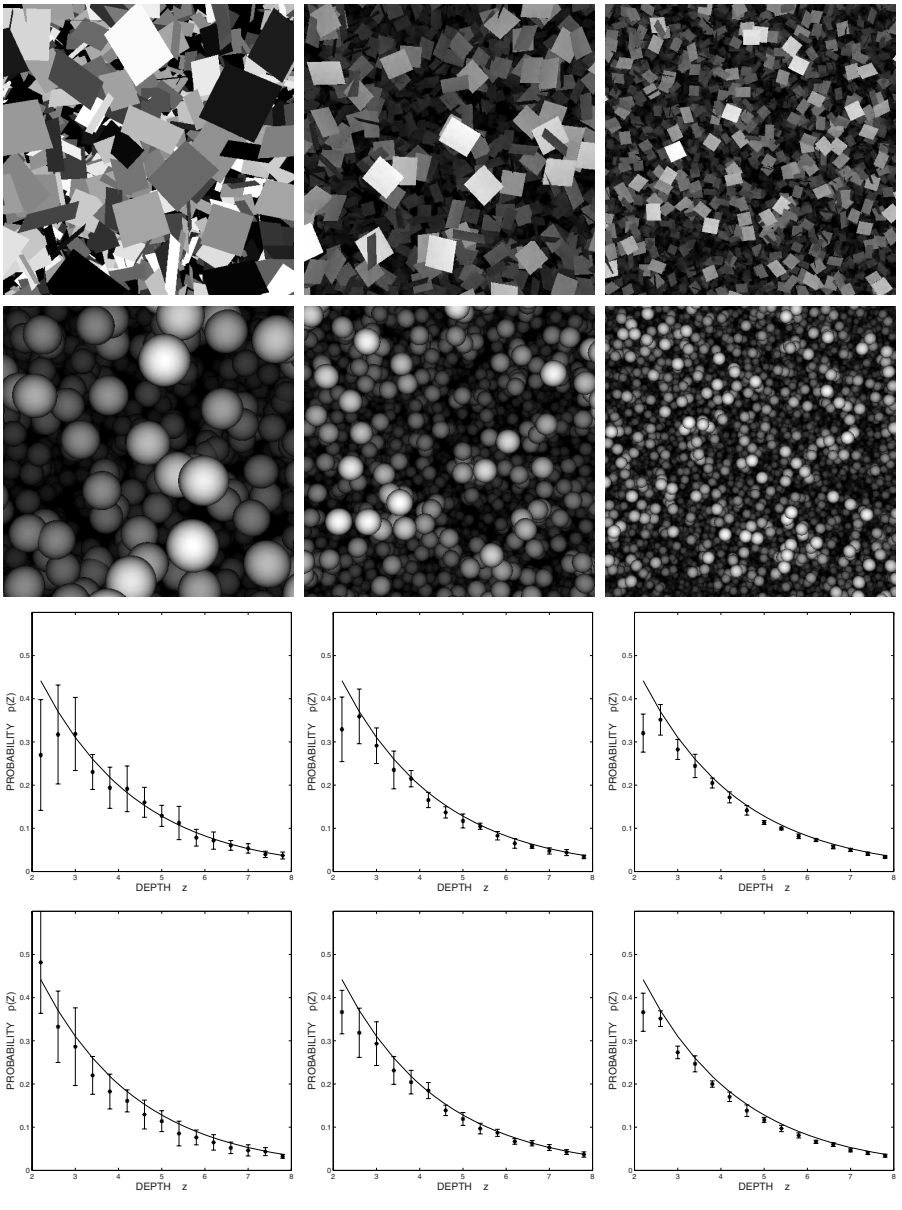


Fig. 1. Examples of rendered scenes consisting of squares or spheres distributed in a view volume. The density and areas of the three scenes have the same λ and so the expected visibility probabilities are the same. Histograms of depths from ten scenes and standard errors in each histogram bin are shown and compared with the theoretical model. The fits are excellent in each case. (See text).

are in the closest depth layer. When the surfaces are large and the density is small, relatively few spheres will contribute to that bin, but each sphere that does contribute will contribute a large number of pixels, hence larger variance.

A second observation is that the theoretical curves slightly overestimate the probabilities, especially at nearest depth bin. The reason for the bias is that the RADIANCE Z buffer gives z values in Euclidean distance from the observer – as do Eqs. (2) and (4). However, as the surfaces are distributed in a rectanguloid whose boundaries are aligned with the viewer coordinate system, there are slightly fewer surfaces at the smallest z values, since for example distance $z = z_0$ falls in the view volume only at one point – where the optical axis meets the near plane. Because the field of view is small, the bias is also small and so we ignore it in the subsequent discussion.

4 Moving Monocular Observer

What happens when the observer moves, in particular, when the observer moves laterally? What can be said about the probability of different image speeds occurring? The key idea is that image speed depends on scene depth and, since we have a probability density on depth, we can convert it to a probability density on image speed.

Assume the projection plane is at depth $z = 1$ unit and the observer's horizontal translation speed is T units per second. The horizontal image speed v_x for a visible surface at depth z is

$$v_x = -\frac{T}{z} \tag{5}$$

and the units are radians per second. Following van Hateren [27], the probability $p(z)\,dz$ that the visible surface lies in the depth interval $[z, z+dz]$ can be related to the probability $p(v_x)\,dv_x$ that a visible surface has image speed in $[v_x, v_x+dv_x]$ as follows. Eq. (5) implies

$$dz = -\frac{T}{v_x^2} dv_x \ . \tag{6}$$

To interpret the minus sign, note that an increase in depth ($dz > 0$) implies a decrease in speed ($dv_x < 0$). Combining Eq. (4) and (6) with

$$p(z)dz = p(v_x)dv_x$$

and defining $v_0 = \frac{T}{z_0}$ to be the image speed for a point on the near clipping plane, we get

$$p(v_x) = \frac{\eta\pi R^2 T}{v_x^2} e^{-\eta(\pi R^2 T(\frac{1}{v_x} - \frac{1}{v_0}))}. \tag{7}$$

Figure 2 shows histograms of the image speeds, plotted from slow to fast (corresponding to depths far to near). Note that the probability density $p(v_x)$ is non-monotonic. While at first blush this may seem to contradict the depth

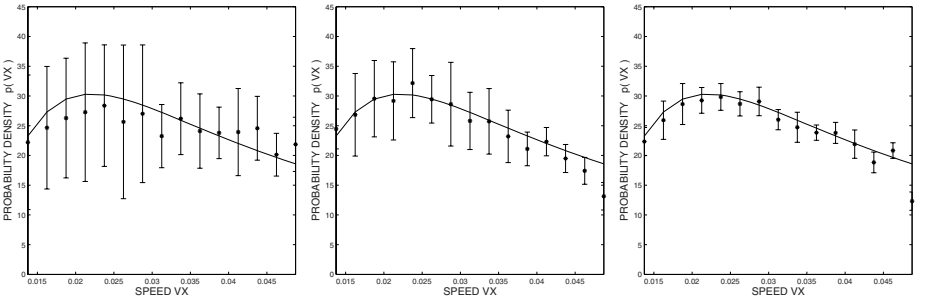
histograms from Fig. 1, in fact it does not. We have binned the velocities in uniform Δv bins, but these do not correspond to the uniform Δz bins.

Again note that the standard errors in the velocity histograms are greater for the plot on the left, which corresponds to large objects (R large). The reason is similar to what was argued for Fig. 1.

One final note is that if we define $v'_x \equiv \frac{v_x}{T}$ then

$$p(v_x)dv_x = \eta\pi R^2(\frac{T}{v_x})^2 e^{-\eta(\pi R^2(\frac{T}{v_x}-\frac{T}{v_0}))} d\frac{v_x}{T} = p(v'_x)dv'_x. \tag{8}$$

Thus the observer speed T merely re-scales the abscissa of the histogram. This is intuitively what we might expect, for example, doubling the observer speed doubles the image speed at each pixel.



$R = 0.1$ $R = 0.05$ $R = 0.025$

Fig. 2. Image speed histograms for the sphere scenes from Figure 1. Standard errors decrease as R decreases i.e. from left plot to right. Data are similar for the square scenes (not shown). The observer is moving at $T = 0.1$ space units per second and so the velocities are in units of radians per second.

5 Binocular Half Occlusions

The velocities discussed in the previous section were instantaneous. When an observer moves over a finite distance in 3D cluttered scenes, however, surfaces can appear from and disappear behind each other at occlusion boundaries [13]. This problem also arises in binocular stereo, of course, and arguably is more severe in stereo since the stereo baseline T is typically larger than the distance moved by a moving observer between frames of a video sequence. In stereo this appearance and disappearance phenomenon is know as *binocular half-occlusion* [1,3]. We now introduce a probability model for half occlusions in 3D cluttered scenes.

Consider the two cylinders C_l and C_r whose axes are the line segments joining some point at depth z to the two eyes (see Appendix). Following similar reasoning to case of spheres in Sec. 3.2, the point at depth z is visible to *both* eyes if and only if *neither* cylinder contains a sphere center. Moreover, the conditional probability that a point at depth z is visible to the right eye, given it is visible to the left eye is:

$$p(z \textit{ visible to right} \mid \textit{visible to left}) = \frac{p(z \textit{ visible to both eyes})}{p(z \textit{ visible to left})}.$$

Recalling Eq. (1) and noting that $vol(C_r \cup C_l) = vol(C_l) + vol(C_r \backslash C_l)$, we get:

$$p(z \textit{ visible to right} \mid \textit{visible to left}) = \frac{e^{-\eta\ vol(C_l \cup C_r)}}{e^{-\eta\ vol(C_l)}} = e^{-\eta\ vol(C_r \backslash C_l)} \qquad (9)$$

We thus need an expression for $vol(C_r \backslash C_l)$. An exact expression is quite complicated since it involves the intersections of two cylinders at a general angle. But using arguments of a similar flavor as those used in [30,14] we can derive the approximation (see Appendix):

$$vol(C_r \backslash C_l) \approx \begin{cases} \pi(\frac{2Rz}{T} - z_0)R^2 + \frac{2z}{T}R^3, & \text{if } \frac{z-z_0}{z} > \frac{2R}{T} \\ \frac{RT(z-z_0)^2}{z}, & \text{otherwise.} \end{cases}$$

This model is examined using computer graphics simulations and the same scenes as before. To compute which points are visible to both eyes, the following trick is used. Treating the viewing position as the left eye, two versions of each scene are rendered: one in which the scene is illuminated by a point source from the viewing position (left eye) and a second in which the scene is illuminated by a point light source at the right eye's position. Then pixels that have non-zero intensity in both rendered images are visible to both eyes. Conditional

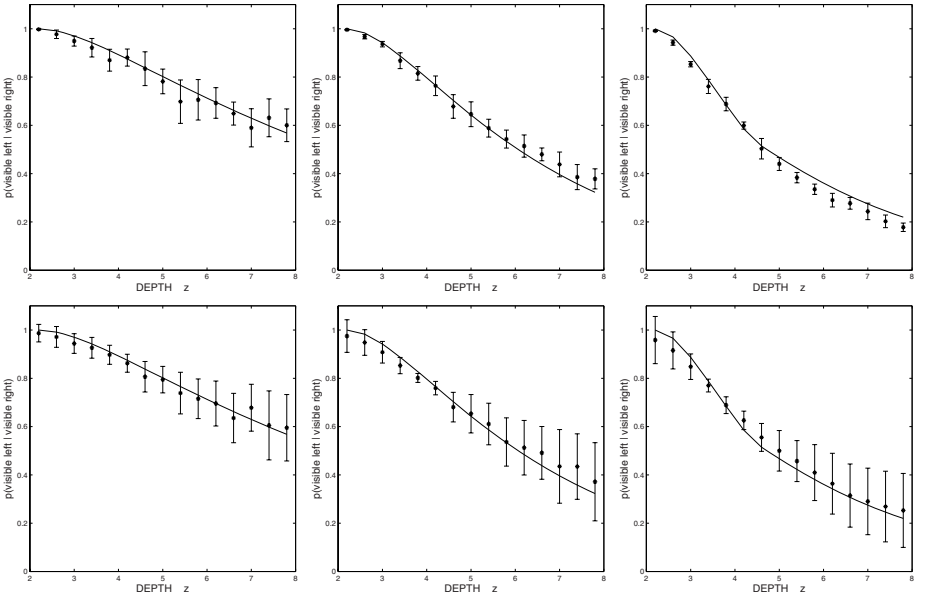


Fig. 3. Conditional probabilities that a surface at depth z is visible to right eye, given it is visible to left eye, for large (left), medium (middle), small (right) scenes

probabilities are estimated from the relative number of pixels at which the surface is visible to both eyes versus the number visible just to the left eye.

Figure 3 compares the model of conditional probabilities with data obtained from RADIANCE simulations. The distance between the eyes was $T = 0.1$ which corresponds to the radius of the spheres in the large sphere scenes. The conditional probabilities are very well fit by the model. While this is not so surprising for the sphere case – it merely verifies that our approximation for $vol(C_r \backslash C_l)$ is good – it is somewhat surprising that the model should produce a good fit for the squares case as well, as the model was derived under the assumption that the scene consists of spheres.

Another interesting observation is that, for both λ and T fixed, the conditional probabilities depend strongly on R, that is, on the size of the surfaces. For the scenes with high-density and small-R objects, the conditional probability falls off faster with z. That is, all other things (z, λ, T) being equal, binocular half occlusions are more common when the scene consists of smaller objects.

6 Limitations

Here we discuss a few limitations of the model, which will be investigated in future work when the model is compared to real imagery. One limitation is the assumption of uniform density and size of objects. For example, the distribution of leaves and branches on typical trees and shrubs is not uniform, *e.g.* leaves tend to clump together near branches and the branch distribution itself is not uniform. This issue has been studied, for example, by plant scientists who are interested how light penetrates a tree canopy [25] and how the penetration depends on geometric and statistical properties of the plant such as leaf area, orientation, and branching factors [16].

A second limitation is that the model ignores partial occlusions which can arise for example from the finite size of pixels. To account for partial occlusion, we would need to modify the model slightly by considering an R-dilated cone rather than an R-dilated ray, such that the solid angle of the cone is that of a pixel. We claim that the difference between the two is insignificant for the case we are considering in which the objects sub-tend a solid angle much greater than that of a single pixel. However, in other cases in which the objects sub-tend a much smaller visual angle (or in which the depth of field is small), we would need to consider partial occlusions. Both of the above limitations are interesting topics to explore in future research.

Acknowledgements

This research was supported by a grant from FQRNT. Thanks to Vincent Couture for helpful discussions, and to Allan Jepson for pointing out the relationship in Eq. (8).

References

1. Belhumeur, P.: A Bayesian approach to binocular stereopsis. International Journal of Computer Vision 19(3), 237–260 (1996)
2. Bravo, M.J., Farid, H.: A scale invariant measure of clutter. Journal of Vision 8(1), 1–9 (2008)
3. Egnal, G., Wildes, R.P.: Detecting binocular half-occlusions: Empirical comparisons of five approaches. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(8), 1127–1133 (2002)
4. Feller, W.: Introduction to Probability Theory and Its Applications. Wiley Series in Probability and Mathematical Statistics, vol. 1 (1968)
5. Garg, K., Nayar, S.K.: Vision and rain. International Journal of Computer Vision 75(1), 3–27 (2007)
6. Grenander, U., Srivastava, A.: Probability models for clutter in natural images. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(4), 424–429 (2001)
7. Hall, P.: Introduction to the Theory of Coverage Processes. John Wiley & Sons, Inc., Chichester (1988)
8. Hegeman, K., Premože, S., Ashikhmin, M., Drettakis, G.: Approximate ambient occlusion for trees. In: I3D 2006: Proceedings of the 2006 symposium on Interactive 3D graphics and games, pp. 87–92. ACM, New York (2006)
9. Ward Larson, G., Shakespeare, R.: Rendering with Radiance: The Art and Science of Lighting Visualization. Morgan Kaufmann, San Francisco (1998)
10. Lee, A.B., Mumford, D., Huang, J.: Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model. International Journal of Computer Vision 41(1/2), 35–59 (2001)
11. Matheron, G.: Random Sets and Integral Geometry. John Wiley and Sons, Chichester (1975)
12. Mittal, A., Davis, L.S.: A general method for sensor planning in multi-sensor systems: Extension to random occlusion. International Journal of Computer Vision 76(1), 31–52 (2008)
13. Mutch, K.M., Thompson, W.B.: Analysis of accretion and deletion at boundaries in dynamic scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence 7(2), 133–138 (1985)
14. Nadler, B., Fibich, G., Lev-Yehudi, S., Cohen-Or, D.: A qualitative and quantitative visibility analysis in urban scenes. Computers and Graphics 23(5), 655–666 (1999)
15. Narasimhan, S.G., Nayar, S.K.: Vision and the atmosphere. International Journal of Computer Vision 48(3), 233–254 (2002)
16. Prusinkiewicz, P.: Modeling of spatial structure and development of plants: a review. Scientia Horticulturae 74, 113–149 (1998)
17. Grzywacz, N.M., Balboa, R.M., Tyler, C.W.: Occlusions contribute to scaling in natural images. Vision Research 41(7), 955–964 (2001)
18. Rosenholtz, R., Li, Y., Nakano, L.: Measuring visual clutter. Journal of Vision 7(2), 1–22 (2007)
19. Roth, S., Black, M.J.: On the spatial statistics of optical flow. International Journal of Computer Vision 74(1), 33–50 (2007)
20. Ruderman, D.L.: Origins of scaling in natural images. Vision Research 37(23), 3385–3398 (1997)

21. Ruderman, D.L., Bialek, W.: Statistics of natural images: scaling in the woods. Physical Review Letters 73, 814–817 (1994)
22. Schechner, Y.Y., Karpel, N.: Clear underwater vision. IEEE Conf. on Computer Vision and Pattern Recognition 1, I–536–I–543 (2004)
23. Serra, J.P.: Image Analysis and Mathematical Morphology. Academic Press, London (1982)
24. Simoncelli, E.P., Adelson, E.H., Heeger, D.J.: Probability distributions of optical flow. In: Proc Conf. on Computer Vision and Pattern Recognition, Mauii, Hawaii, pp. 310–315. IEEE Computer Society Press, Los Alamitos (1991)
25. Sinoquet, H., Sonohat, G., Phattaralerphong, J., Godin, C.: Foliage randomness and light interception in 3d digitized trees: an analysis of 3d discretization of the canopy. Plant Cell and Environment 29, 1158–1170 (2005)
26. Szeliski, R., Golland, P.: Stereo matching with transparency and matting. International Journal of Computer Vision 32(1), 45–61 (1999)
27. van Hateren, J.H.: Theoretical predictions of spatiotemporal receptive fields of fly LMCs, and experimental validation. Journal of Comparative Physiology A 171, 157–170 (1992)
28. Weiss, Y.: Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 520–526 (1997)
29. Weiss, Y., Fleet, D.J.: Probabilistic Models of the Brain: Perception and Neural Function. In: Velocity likelihoods in biological and machine vision, pp. 77–96. MIT Press, Cambridge (2002)
30. Zacks, S.: Stochastic Visibility in Random Fields. Lecture Notes in Statistics 95. Springer, Heidelberg (1994)

Appendix

Figures 4 and 5 illustrate two configurations that must be considered for us to calculate $vol(C_l \backslash C_r)$. In both cases, I only need to consider the parts of the cylinder that are beyond z_0 since by definition no sphere centers are present for $z < z_0$. The two cases are distinguished by whether the cross sections of the cylinders (disks) overlap at depth z_0.

If the two cylinders overlap at depth z_0, then we approximate $C_l \backslash C_r$ to be the new volume swept out by a cylinder of radius R if a viewer were to move continuously from the left to right eye's position.[3] By inspection this swept volume is $\frac{RT(z-z_0)2}{z}$, where the factor $\frac{(z-z_0)T}{z}$ is the projection of the interocular distance T to the z_0 plane.

If the two cylinders do not overlap at depth z_0, then there must be a minimal depth z_1 where the two cylinders overlap such that $z_0 < z_1 < z$. This is the situation shown below. In this case we approximate $C_l \cup C_r$ by partitioning $C_l \backslash C_r$ into two sets, namely the points in front of and beyond this depth z_1, respectively. For the points in front of z_1, the volume within $C_l \backslash C_r$ is $\pi R^2(z_1 - z_0)$. Using a similar triangle argument, we note $\frac{z-z_1}{2R} = \frac{Z}{T}$, and with a little manipulation

[3] This swept volume is a slightly overestimate of $C_l \cup C_r$. It assumes the point at depth z, which lies on the axis of the cylinders, is seen by *all* viewpoints in between the left and right eye's position.

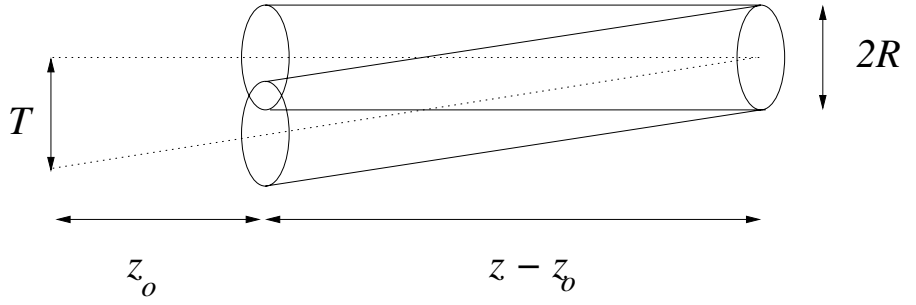


Fig. 4. A point at depth z is visible to two eyes separated by a distance T if no sphere centers lie in the cylinders whose axes join the point to the two eyes. Here we consider the case that the two cylinders overlap at the clipping plane $z = z_0$.

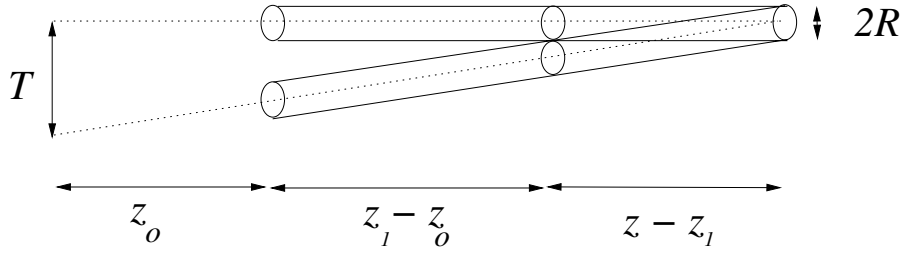


Fig. 5. Similar to Fig. 4 but now we consider the case that the two cylinders do not overlap at the clipping plane $z = z_0$, and that the overlap occurs starting at depth $z = z_1 > z_0$

we can rewrite this volume as $\frac{2R^3Z}{T}$. For the points beyond z_1, we approximate the volume within $C_l \backslash C_r$ using a similar idea as the case of Fig. 4. Now the two cylinders are exactly tangent to each other at depth z_1 so the distance between their axes at this depth is $2R$. we approximate the volume of $C_l \backslash C_r$ that lies beyond z_1 as the volume swept out by the cylinder, namely $\frac{2R}{2}(z - z_0)$. This is a slight overestimate of the volume i.e. an approximation.

A Generative Shape Regularization Model for Robust Face Alignment

Leon Gu and Takeo Kanade

Computer Science Department
Carnegie Mellon University

Abstract. In this paper, we present a robust face alignment system that is capable of dealing with *exaggerating expressions*, *large occlusions*, and *a wide variety of image noises*. The robustness comes from our shape regularization model, which incorporates constrained nonlinear shape prior, geometric transformation, and likelihood of multiple candidate landmarks in a three-layered generative model. The inference algorithm iteratively examines the best candidate positions and updates face shape and pose. This model can effectively recover sufficient shape details from very noisy observations. We demonstrate the performance of this approach on two public domain databases and a large collection of real-world face photographs.

1 Introduction

Face alignment is a challenging problem especially when it comes to real-world images. The main difficulty arises from pervasive ambiguities in low-level image features. Consider the examples in Figure 1. While the main face structures are present in the feature maps, the boundaries of face components are frequently disrupted by gaps or corrupted by spurious fragments. Strong gradient responses could be due to reflectance, occlusion, image blur, or fine facial textures. In contrast, the boundaries of nose, jawline, and other subparts could be obscure or even barely perceptible. Detecting individual facial landmarks in such feature maps often yields noisy results. Yet, an interesting question is, on the basis of such "observations" what is the best hypothesis that one can make?

We believe that the ability to recover sufficient shape details from noisy image observations lies at the core of a *robust* face alignment system. In this paper, we address this problem by use of a three-layered generative model. The model allows multiple candidate positions to be generated for each facial landmark, and the image likelihood of seeing a landmark at one particular position is defined at the bottom layer of the model. The best candidate is treated as a hidden variable to be decided. The middle layer models global geometric transformation in which a noise term is assigned to each landmark to measure its fitting error. The top layer models prior shape distribution as Gaussian mixture, in which the number of free parameters in each Gaussian component is restricted to be small. This prior is compact, and also flexible to capture large shape deformations. We use expectation-maximization algorithm to iteratively select candidates and estimate

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 413–426, 2008.

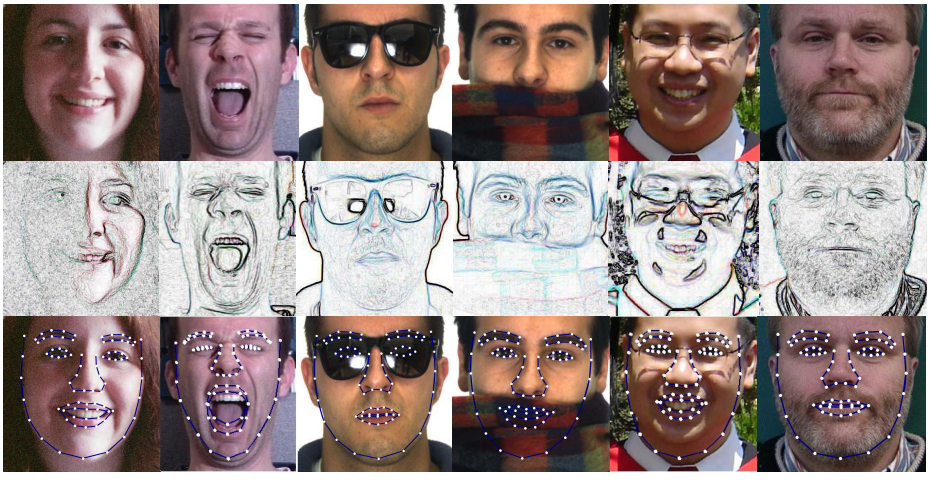


Fig. 1. Real-world face images could be extremely noisy in the eyes of computers. As shown in in the gradient feature maps (second row), face topologies could be significantly corrupted due to various kinds of factors. However, our model can still recover sufficient shape details from such noisy observations while using a simple gradient-based landmark detector. The third row shows our alignment results.

face shape and pose. We show that the model can deal with observation noises in a way such that the major trends of shape deformations can still be recovered, and the well-matched facial landmarks can be differentiated from outliers. One advantage of putting deformable matching in this generative setting and using EM for inference is that our treatment is guaranteed to be consistent.

The rest of this paper is organized as follows. The background and related works are introduced in section 2. We explain the details of the model including the inference algorithm in section 3, and show the experimental results and comparison in section 4. We conclude in section 5.

2 Background

Leveraging prior knowledge on low-level image interpretation has been the core idea of deformable matching since Eighties [1] [2] [3] [4] [5]. The forms of priors that have been imposed on images are diverse, including generic properties of parametric curves such as continuity and smoothness [1], specific object structures defined by assemblies of flexible curves [3], linear shape deformation subspace learned from landmarked training images [1], and shape dynamics [5]. Latter developments have been focused on modeling object appearance [6] [7]; exploiting nonlinear [8] [9] [10] and 3D [7] [11] [12] shape prior; refining alignment results using Markov network [13]; and seeking efficient matching algorithms by dynamic programming [14] [15] or belief propagation [16] when shape priors are

defined on discrete spaces. Progresses have also been made to improve facial landmark encoding [17] and detection [18].

It is also worth mentioning that Gaussian mixture was used in [8] to refine the linear subspace model [4], but their shape regularization approach is completely deterministic. Our generative formulation for deformable matching is similar to the recently proposed methods by Zhou et al. [19] and Gu et al. [11]. These methods, however, are restricted to linear shape deformations, and likely to oversimplify shape observations - only one candidate position for each facial landmark.

3 The Generative Model for Shape Regularization

We follow the standard shape representation. A face Q consists of N landmark points, i.e., $Q = (Q_1^x, Q_1^y, \ldots, Q_N^x, Q_N^y)^t$. The landmarks are commonly placed along the boundaries of face components. The geometry information of Q decouples into two parts: a canonical shape S, and a rigid transformation $\mathcal{T}_\theta$ that maps S from a common reference frame to the coordinate plane of the target image I.

When low-level features are ambiguous, it makes sense to allow landmark detectors to produce multiple candidates. Suppose that for the n-th landmark, there are K candidate positions $Q_{nk} = (Q_{nk}^x, Q_{nk}^y)$ located on the image. Let $\mathcal{Q} = \{Q_{nk}\}$ denote the whole set of $N \times K$ candidates. Our goal is to estimate the shape S and the pose θ from $\mathcal{Q}$.

3.1 Model Structure

First a decision needs to be made for each landmark to select the "best" candidate. We introduce a latent variable h that assigns one candidate position to each landmark. h is a binary $N \times K$ matrix, in which each row contains only one "1" and all other entries are zeros. The image likelihood of seeing a landmark at one particular position Q_{nk} is measured by

$$p(I|h_{nk} = 1) = p(I|Q_{nk}) = \pi_{nk}, \tag{1}$$

and subject to the constraint $\sum_k \pi_{nk} = 1$. Let $\mathcal{Q}(h)$ denote the set of positions selected by h. Then we can write the n-th point of $\mathcal{Q}(h)$ as

$$\mathcal{Q}_n(h) = \left(\sum_{k=1}^{K} h_{nk} Q_{nk}^x, \sum_{k=1}^{K} h_{nk} Q_{nk}^y \right)^t. \tag{2}$$

Assume that $\mathcal{Q}(h)$ is generated from the canonical shape S and the pose θ by first transforming S according to θ, then adding an independent Gaussian noise to each landmark. We can write the conditional probability of $\mathcal{Q}(h)$ as

$$p(\mathcal{Q}(h)|S, \theta) = \mathcal{N}(\mathcal{T}(S, \theta); \Sigma), \tag{3}$$

where the covariance matrix Σ is diagonal, i.e., $\Sigma = \text{diag}(\rho_1, \rho_1, \ldots, \rho_N, \rho_N)$. The independence assumption is valid because the detection of landmarks is often performed independently to each other. The variance ρ_n measures the noise level of the observation $\mathcal{Q}_n(h)$. A non-informative prior is put on the similarity transformation parameters $\theta = \{R, s, t\}$ to allow arbitrary rotation, translation and isotropic scaling.

We define the prior distribution over the shape S as a mixture of constrained Gaussian [20],

$$p(S|b) = \sum_{l=1}^{L} \pi_l \mathcal{N}(\Phi_l b_l + \mu_l; \sigma_l^2 I), \tag{4}$$

where $b = \{b_l\}$ denotes the deformation coefficients, and the model parameters associated with each Gaussian component are the mixing rate π_l, the linear principal subspace spanned by the columns of Φ_l, the mean shape μ_l, and the isotropic shape noise with zero mean and variance $\sigma_l^2 I$. Compared to general Gaussian mixtures, this mixture model is more compact and contains less free parameters. Each Gaussian component is restricted in a linear subspace spanned by Φ_l, and the dimension of the subspace is decided by the percentage of shape variance that is desired to be preserved. Within each subspace, b_l controls the amount of shape deformation, and σ_l^2 determines the variance of shape noise. The variance σ_l^2 is computed as the average residual shape variance outside of the subspace,

$$\sigma_l^2 = \frac{1}{N - M_l} \sum_{m=M_l+1}^{N} \lambda_{lm}. \tag{5}$$

Here $\{\lambda_{lm}\}$ denote the eigenvalues, which are arranged in a deceasing order, and M_l is the subspace dimension. Other model parameters $\{\pi_l, \mu_l, \Phi_l\}$ are also learned from training shapes (see [20] for details). The difference from [20] is that we model the deformation prior $p(b_l)$ as a diagonal Gaussian

$$p(b_l) = \mathcal{N}(0; Diag(\lambda_{l1}, \ldots, \lambda_{lM_l})), \tag{6}$$

and restrict the columns of matrix Φ_l to be orthogonal. We then rewrite (4) by introducing a latent component label z,

$$p(S|b, z) = \prod_{l} \mathcal{N}^{z_l}(\Phi_l b_l + \mu_l; \sigma_l^2 I), \tag{7}$$

and putting a multinomial distribution on z

$$p(z_l = 1) = \pi_l. \tag{8}$$

Combining (1) $\sim$ (8), we construct a hierarchical deformable model.

3.2 The Alignment Algorithm

Our problem now is to estimate the *deformation parameters* b and the *transformation parameters* θ from the candidate points set $\mathcal{Q}$. This is formulated as a

MAP problem, i.e., finding the optimum $\{b^*, \theta^*\}$ by maximizing the posterior $p(b, \theta|\mathcal{Q})$, and solved by EM. First we look at the joint distribution over the assignment variable h, the mixture component label z, the hidden shape vector S, the deformation parameters b, and the transformation parameters θ,

$$p(b,\theta,S,h,z|I) \propto p(S|b,z)p(b)p(z)p(\mathcal{Q}(h)|S,\theta)p(I|h) \tag{9}$$

Taking the expectation $\langle\cdot\rangle$ of the log of (9) over the posterior of the latent variables S, h, z, we obtain the so-called Q-function,

$$\begin{aligned}\langle \log p(b,\theta,S,h,z|I)\rangle \propto \langle \log p(S|b,z)\rangle + \log p(b) \\ + \langle \log p(z)\rangle + \langle \log p(\mathcal{Q}(h)|S,\theta)\rangle + \langle \log p(I|h)\rangle\,.\end{aligned} \tag{10}$$

In the E-step, we compute the sufficient statistics that are required to evaluate (10); and in the M-step we maximize (10) to find the updated shape and pose.

Expectation Step. Substituting for the expectations on the right-hand side of (10) and absorbing terms that are independent of b and θ into an additive constant, we expand (10) as

$$\begin{aligned}&\langle \log p(b,\theta,S,h,z|\mathcal{Q})\rangle_{S,h} \propto \sum_{n,k}\langle h_{nk}\rangle \log \pi_{nk} \\ &+\sum_{l}\langle z_l\rangle \log \pi_l - \tfrac{1}{2}\sum_{n}\rho_n^{-2}\left\langle \|\mathcal{Q}(h_n) - \mathcal{T}(S_n,\theta)\|^2\right\rangle \\ &-\tfrac{1}{2}\sum_{l}\left\langle z_l\sigma_l^{-2}\|S-\Phi_l b_l-\mu_l\|^2\right\rangle - \tfrac{1}{2}\sum_{l,m}\lambda_{lm}b_{lm}^2.\end{aligned} \tag{11}$$

In order to evaluate (11), we need the joint and marginal posteriors of all discrete and continuous latent variables. We first write down the joint posterior, which is given by

$$\begin{aligned}p(S,h,z|I,b,\theta) \propto \\ p(S|b,z)p(z)p(\mathcal{Q}(h)|S,\theta)p(I|h).\end{aligned} \tag{12}$$

Because of the conditional independence assumptions made in (1) (3) (7), the right side of (12) can be further factorized between individual points

$$p(z)\prod_{n=1}^{N}\left\{p(S_n|b,z)p(\mathcal{Q}(h_n)|S_n,\theta)p(I|h_n)\right\}. \tag{13}$$

Thus we can evaluate $p(S_n, h_n|I, b, \theta, z)$ for each point separately. We factorize it by chain rule

$$p(S_n,h_n|I,b,\theta,z) = p(S_n|h_n,I,b,\theta,z)p(h_n|I,b,\theta,z). \tag{14}$$

According to the Bayes' rule, the first factor decomposes into a product of the prior $p(S_n|b,z)$ and the likelihood $p(\mathcal{Q}(h_n)|S_n,\theta)$. Because both distributions are Gaussian and similarity transform is linear, the posterior is still a Gaussian.

$$p(S_n|h_n,l,b,\theta,z) = \mathcal{N}(\bar{S}_{nkl}, c_{nl}^2 I_{2\times 2}). \tag{15}$$

Its mean and covariance are given by

$$\bar{S}_{nkl} = w_l^1 S_n(b_l) + w_l^2 \mathcal{T}^{-1}(h_n) \tag{16}$$

$$c_{nl}^2 = (\sigma_l^{-2} + s^2 \rho_n^{-2})^{-1} \tag{17}$$

where we have defined

$$w_l^1 = \frac{\sigma_l^{-2}}{\sigma_l^{-2} + s^2 \rho_n^{-2}} \tag{18}$$

$$w_l^2 = \frac{s^2 \rho_n^{-2}}{\sigma_l^{-2} + s^2 \rho_n^{-2}} \tag{19}$$

$$S(b_l) = \Phi_l b_l + \mu_l \tag{20}$$

$$\mathcal{T}^{-1}(h_n) = \mathcal{T}^{-1}(\mathcal{Q}(h_n), \theta) \tag{21}$$

where $\mathcal{T}^{-1}$ denotes the inverse similarity transformation, and the subscript in S_n denotes the n-th landmark. For the second factor in (14) we marginalize the joint posterior $p(S_n, h_n | I, b, \theta, z)$ over S_n

$$\begin{aligned} p(h_n | I, b, \theta, z) &\propto p(I | h_n) \\ &\int p(S_n | b, z) p(\mathcal{Q}(h_n) | S_n, \theta) dS_n. \end{aligned} \tag{22}$$

The integral in (22) is a function of h_n, and its value measures a scaled distance between the model prediction $S(b_z)$ and the observed candidate position specified by h. Requiring that the distribution (22) be normalized, we obtain,

$$p(h_{nk} = 1 | I, b, \theta, z_l = 1) \propto \pi_{nk} r_{nkl} \tag{23}$$

where r_{nkl} is the exponential of the scaled distance, given by

$$r_{nkl} = \exp\{-\frac{||\sigma_l^{-2} S_n(b_l) - s^2 \rho_n^{-2} \mathcal{T}^{-1}(h_{nk})||^2}{2(\sigma_l^{-2} + s^{-2} \rho_n^2)(s/\rho_n)^8}\} \tag{24}$$

From (15) and (23), we shall see that the distribution $p(S_n | b, \theta, z)$ is a mixture of K Gaussian components, in which the mixing rate is given by (23) and the component mean and covariance are given by (16) and (17) respectively. We now integrate (13) over both S and h, and make use of (23) to compute $p(z | I, b, \theta)$

$$p(z_l = 1 | I, b, \theta) \propto \pi_l \prod_n \sum_k \pi_{nk} r_{nkl}. \tag{25}$$

Further decomposing the right-hand side of (11), we shall find this expectation depends on the posterior distributions only through the following statistics $\langle z_l \rangle$, $\langle h_{nk} \rangle$, $\langle S_n \rangle$, $\langle S_n^t S_n \rangle$, $\langle h_{nk} Q_{nk}^t R S_n \rangle$, $\langle z_l S_n \rangle$, $\langle z_l S_n^t S_n \rangle$. At this point, we shall find it convenient to define

$$\pi_l' := p(z_l = 1 | I, b, \theta) \tag{26}$$

$$\pi_{nkl}' := p(h_{nk} = 1 | I, b, \theta, z_l = 1) \tag{27}$$

$$\bar{S}_{nl} := \sum_k \pi_{nkl}' \bar{S}_{nkl}. \tag{28}$$

The sufficient statistics are easily evaluated from the distributions defined by (15) (23) (25), to give

$$\langle z_l \rangle = \pi'_l \tag{29}$$

$$\langle h_{nk} \rangle = \sum_l \pi'_l \pi'_{nkl} \tag{30}$$

$$\langle S_n \rangle = \sum_l \pi'_l \bar{S}_{nl} \tag{31}$$

$$\langle z_l S_n \rangle = \pi'_l \bar{S}_{nl} \tag{32}$$

$$\left\langle S_n^t S_n \right\rangle = \sum_{k,l} \pi'_l \pi'_{nkl} (\bar{S}_{nkl}^t \bar{S}_{nkl} + 2c_{nl}^2) \tag{33}$$

$$\left\langle h_{nk} Q_{nk}^t R S_n \right\rangle = Q_{nk}^t R \sum_l \pi'_l \pi'_{nkl} \bar{S}_{nkl} \tag{34}$$

$$\left\langle z_l S_n^t S_n \right\rangle = \pi'_l \sum_k \pi'_{nkl} (\bar{S}_{nkl}^t \bar{S}_{nkl} + 2c_{nl}^2) \tag{35}$$

Thus in E step we use old parameters $\{b^{\text{old}}, \theta^{\text{old}}\}$ to evaluate the posterior statistics (29∼35).

Maximization Step. In the M step, we maximize (11) with respect to b and θ using the sufficient statistics (29 ∼ 35). Note that b and θ are decoupled in (11), thus we can solve them separately. Taking the derivative of (11) with respect to b_l and setting it to zero and making use of the statistics (32) (29), we obtain the updating equation for the deformation parameters,

$$\tilde{b}_l = \frac{\langle z_l \rangle \, \sigma_l^{-2} (\Phi_l^t \bar{S}_l - \mu_l)}{\langle z_l \rangle \, \sigma_l^{-2} + \Lambda_l^{-1}}, \tag{36}$$

where $\bar{S}_l$ is a shape vector defined by $\bar{S}_l = (\bar{S}_{1l}, \ldots, \bar{S}_{Nl})^t$ and $\bar{S}_{nl}$ is defined in (28).

Substituting (30∼34) into (11), taking the derivatives with respect to s and t and setting them to zero, we obtain the updating equations for translation and scale,

$$\tilde{t} = \frac{1}{N} \sum_{n,k,l} \pi'_l \pi'_{nkl} Q_{nk} \tag{37}$$

$$\tilde{s} = \frac{\sum_n \left(\tilde{t}^t \tilde{R} \langle S_n \rangle - \sum_k \left\langle h_{nk} Q_{nk}^t \tilde{R} S_n \right\rangle \right)}{\sum_n \langle (S_n^t S_n) \rangle}. \tag{38}$$

For rotation, maximizing (11) is equivalent to maximizing the trace

$$\text{Trace} \left\{ R \sum_n \left(\langle S_n \rangle \, \tilde{t}^t - \langle h_k S_n Q_{mk}^t \rangle \right) \right\}. \tag{39}$$

Therefore, the optimal rotation $\tilde{R}$ can be computed by polar decomposing the matrix $\sum_n \left(\langle S_n \rangle \, \tilde{t}^t - \langle h_k S_n Q_{mk}^t \rangle \right)$.

3.3 Analysis

We first summarize the alignment algorithm. The inputs of the algorithm are the candidate position set $\{Q_{nk}\}$ and the image likelihood of each candidate $\{\pi_{nk}\}$; and the outputs are the optimal deformation and pose parameters $b = \{b_l\}, \theta = \{R, s, t\}$. Starting from an initial estimate $\{b^0, \theta^0\}$, the algorithm first computes a set of sufficient statistics (29) $\sim$ (35), then use them to update b (36) and θ (37) $\sim$ (39). Next we analyze the algorithm by looking into the details in (29) $\sim$ (39).

Making Use of Multiple Candidates: the selection of the best candidate is performed in a "soft" way by evaluating the posterior assignment probabilities (23) (30). Conditional on a particular mixture component l, the prior assignment π_{nk} is modulated by the similarity measure r_{nkl} between the position $S_n(b_l)$ predicted from the l-th component and the observed position Q_{nk}, to produce the conditional posterior assignment (23); the marginal posterior (30) is then computed by averaging (23) over all mixture components. These posteriors are used to weight candidate points to generate "averaged" observations in (28) - for a particular mixture component l, and (31) - for the whole mixture distribution. The posterior assignments are also used in pose estimation through (37) $\sim$ (39), to increase the contribution of good candidates.

Regularization by Multi-modal Prior: The averaged "observation" (28) is regularized by shape priors to produce a shape estimate. This regularization step is performed in (36), by first computing the subspace representation $\bar{b}_l = (\Phi_l^t \bar{S}_l - \mu_l)$ in each mixture component, then shrinking the deformation coefficients $\bar{b}_{lm}$ by a factor of $\frac{\langle z_l \rangle \sigma_l^{-2}}{\langle z_l \rangle \sigma_l^{-2} + \lambda_{lm}^{-1}}$. We see that the *degree of regularization* is determined in terms of three factors: the subspace responsibility $\langle z_l \rangle$, the shape variance λ_{lm} and the shape noise variance σ_l:

1. (between subspaces): a smaller $\langle z_l \rangle$, meaning that the observed shape is less likely to be generated from the l-th subspace, leads to a heavier regularization on b_l;
2. (within a subspace): a smaller shape variance λ_{lm} leads to a heavier penalization on the corresponding deformation component $\bar{b}_{lm}$ and vice versa.
3. (overall): the overall degree of regularization is controlled by the variance of shape noise σ_l^2 (5), which is determined by the percentage of shape variance that is preserve in each subspace. Reducing the percentage leads to larger regularization on b.

Identifying Outliers: the resistance to outliers is achieved by the observation noise model (3). Because observation noises are unpredictable, its variance ρ_n is unlikely to be learned as a prior. Therefore we set the initial ρ_n to be same for all landmark points, then change it according to the fitting error between the model prediction and the averaged candidate position

$$\rho_n = c \| \mathcal{T}(S_n(\tilde{b}), \tilde{\theta}) - \sum_k \pi_{nk} Q_{nk} \|, \tag{40}$$

where c is a constant. We update ρ_n whenever there exist new landmark detection results (see section 3.4), and used it to compute the weights (18) (19). These weights are in turn served for penalizing outliers (16) in both shape and pose estimation steps.

3.4 Initialization and Landmark Detection

The alignment program is initialized by a rotation-invariant face detector [21], which scans a target image and produces the initial guess of the pose $\theta^{(0)}$. The initial deformation parameter $b^{(0)}$ is simply set to be zero. The average training shape $\sum_l \pi_l \mu_l$ is transformed by $\theta^{(0)}$ and superimposed on the image.

We construct a simple gradient based landmark detector in a similar way to [4]. The neighboring image gradient centered on each landmark is normalized (L_1-distance equals to one) and modeled as a multivariate Gaussian whose mean and variance are learned from training data. For each landmark we search its nearby region and measure the probability of each pixel. That produces a response map, and the K largest local modes found in the map are used as the candidate positions. The response score of these candidates are further normalized and used as the image likelihood π_{nk}. This landmark detection procedure and the shape inference algorithm (29) $\sim$ (39) are performed recursively on a Gaussian image pyramid from the coarsest level to the finest.

4 Experiments

We first evaluated the proposed face alignment method on manually labeled frontal face images. The images are collected from two sources: CMU Multi-PIE database [22] and AR database [23]. All faces are frontal and non-occluded. Although captured in a controlled environment, this dataset covers large variations on subject identity, expression and illumination. We also extensively tested our program on a large collection of *real-world* images. Our goal in these experiments is to show that by putting deformable matching in our framework, we can improve the alignment accuracy, and more importantly, deal with a wide variety of situations in real-world face photographies.

Mis-Alignment Error When Images Are Clean. We compared our face alignment program with two previous techniques, namely, Active Appearance Model [6] and Bayesian Tangent Shape Model [19] on the labeled dataset. 800 randomly selected images are used for training, the rest 480 images are used for testing. For a fair comparison, same landmark detector is applied in all three methods; the initial shape parameters are set to zero; and the initial poses are generated by first computing a pose from ground-truth landmarks via procrustes analysis, then adding a small random permutation by translation ($-10\% \sim 10\%$ of the face size), rotation ($-15^\circ \sim 15^\circ$) and scaling ($0.9 \sim 1.1$). When computing the average mis-alignment error, we normalize the width of each testing face to be 120 pixels, keeping the aspect ratio unchanged and scaling the height accordingly. The overall mis-alignment error is 7.88 pixels for AAM, 5.90 pixels for BTSM

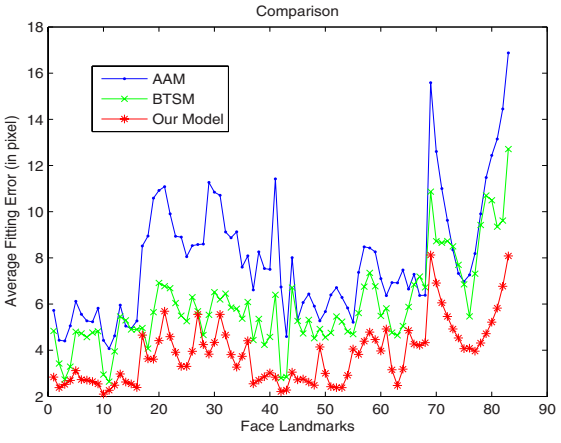


Fig. 2. We evaluate the performance of our model and compare it with AAM [6] and BTSM [19]. The graph plots the average fitting errors for every point: blue for AAM, green for BTSM, and red for our model. The landmarks represent left/right eyes ($1 \sim 8$; $9 \sim 16$), left/right eyebrows ($17 \sim 26$; $27 \sim 36$), nose ($37 \sim 48$), mouth ($49 \sim 68$) and silhouette ($69 \sim 83$) respectively.

and 3.49 pixels for our model. Figure 2 compares the errors obtained by the three methods for each individual point. Our model consistently outperforms the other two techniques on all points.

Occlusions. In the presence of image clutters or occlusions, we expect that our model identifies the noisy points in the fitting process by measuring the discrepancy between model prediction and low-level image observation. Figure 3 shows a few partially occluded face images. Our model can deal with these cases, while traditional deformable models could easily fail. We plot the the estimated shape noise level (40) associated every point. When the points are occluded, such as those along eye contours (at the top row) or along mouth and jaw-line (at the bottom row), the corresponding observations are clearly considered as more ambiguous by the model. Also note that the predictions on the visible part are stable. As a result, small weights are assigned on the occluded points, and their positions are "hallucinated" by combining the information from reliable observations and shape priors according to the penalization rules (16)(31).

Facial Expression. The training images cover six types of expressions including neutral expression, smile, squint, surprise, disgust and scream. By use of a mixture shape prior, our deformable model is capable of capturing a larger range of expression variations than linear models. And the associated shape regularization rule ensures that the model can smooth out a noisy shape observation, and preserve dominant shape deformations. We select the number of mixture components as $L = 3$. Larger L does not improve the alignment performance but increases computational cost. The top row of Figure 4 shows the results on a

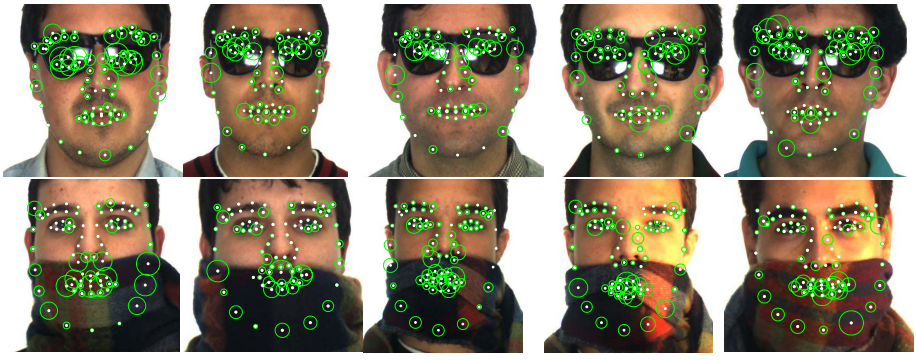


Fig. 3. Our model automatically distinguishes good landmarks from bad ones in the model matching process. White dots represent the alignment results; green circles represent the noise level of each landmark at the end of matching process. Larger observation variance implies a small contribution to the final estimates of shape and pose components.

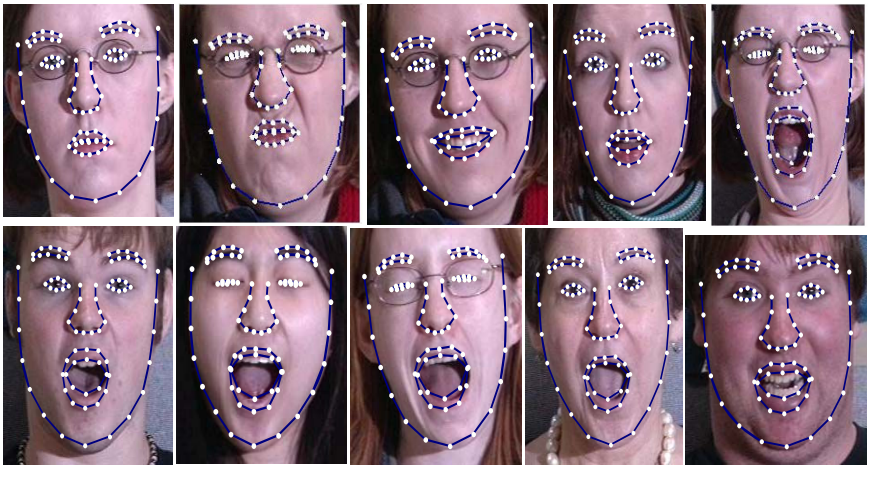


Fig. 4. Our model is capable of dealing with large expression changes. The top row shows the faces of a subject recorded in five different expressions: neutral, disgust, smile, surprise and scream; the bottom row highlights our alignment results on screaming faces.

testing subject in Multi-PIE dataset with different expressions, and the bottom row shows the results on a few images in our second dataset.

In-plane Rotation. The face detector is applied at 8 different orientations in increments of 35°. Our deformable model is capable of dealing with rotation in the range from $-25°$ to 25° degrees. Combining the model with the face detector our alignment program is capable of dealing with full $0° \sim 360°$ in-plane as shown in Figure 5.

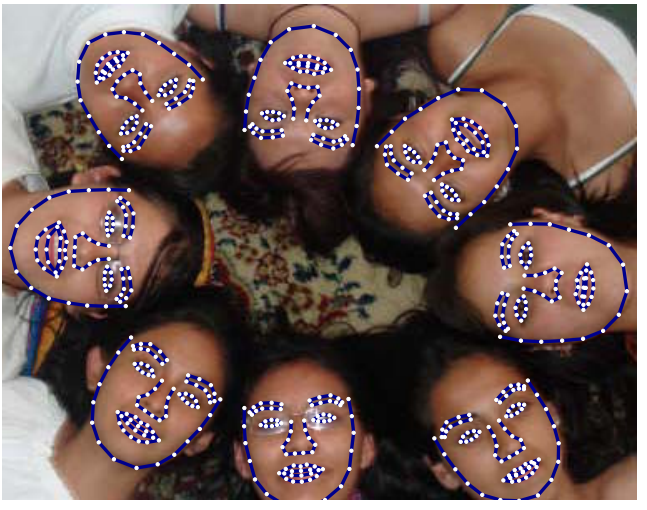


Fig. 5. Combining our deformable model with a rotation-invariant face detector our system is capable of dealing with the full 360 degree in-plane rotation

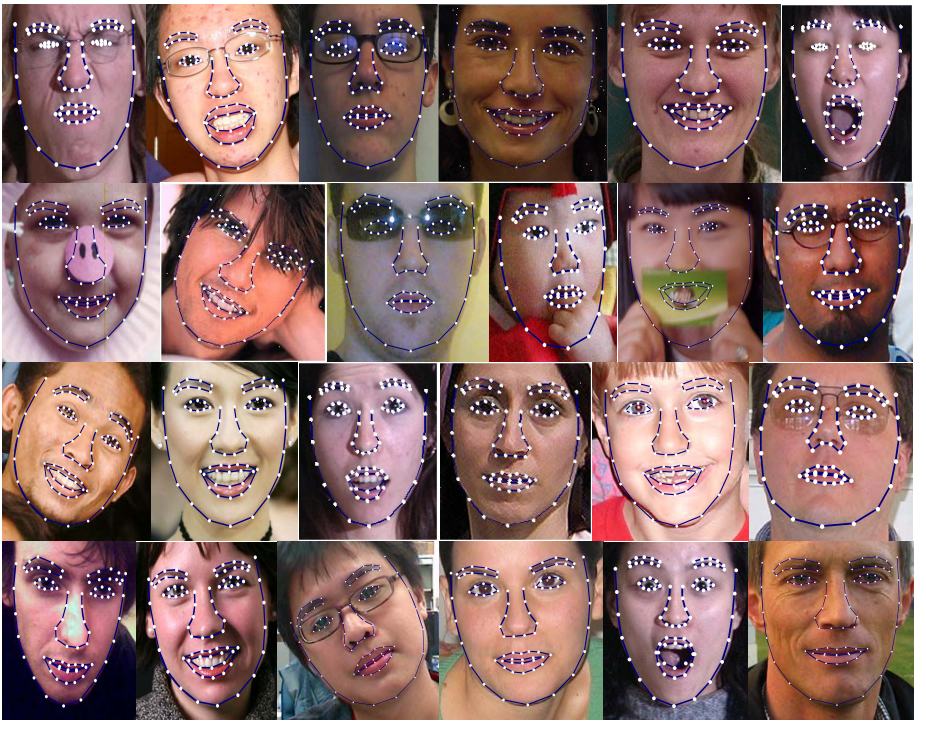


Fig. 6. The output of our face alignment program on some real-world photographs collected from Internet. More results are available on our website.

Real-World Photographs. Figure 6 shows the alignment results on a few real-world face image collected from Internet. More results are available on our website.

5 Conclusion and Discussion

In this paper, we have proposed a new approach for shape regularization by use of a multi-level generative model, and demonstrated its application in face alignment. We show our alignment system is capable of dealing with real-world images with a wide range of imaging conditions and appearance variations. Our model uses image gradients as the only low-level cues. By incorporating the model with other image cues or landmark detectors can potentially further improve the alignment accuracy.

References

1. Terzopoulos, D., Witkin, A., Kass, M.: Snakes: Active contour models. In: International Conference on Computer Vision, pp. 259–268 (1987)
2. Grenander, U., Chow, Y., Keenan, D.M.: Hands: a pattern theoretic study of biological shapes. Springer, New York (1991)
3. Yuille, A.L., Hallinan, P.W., Cohen, D.S.: Feature extraction from faces using deformable templates. International Journal of Computer Vision 8, 99–111 (1992)
4. Cootes, T.F., Taylor, C., Cooper, D., Graham, J.: Active shape models - their training and their applications. Computer Vision and Image Understanding (1995)
5. Isard, M., Blake, A.: Condensation - conditional density propagation for visual tracking. International Journal of Computer Vision 29, 5–28 (1998)
6. Cootes, T., Edwards, G., Taylor, C.: Active appearance models. IEEE Trans. Pattern Anal. Mach. Intell. 23, 681–685 (2001)
7. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d-faces. In: ACM SIGGRAPH (1999)
8. Cootes, T., Taylor, C.: A mixture model for representing shape variation. IVC 17, 567–573 (1999)
9. Twining, C., Taylor, C.: Kernel principal component analysis and the construction of non-linear active shape models. In: British Machine Vision Conference (2001)
10. Zhou, Y., Zhang, W., Tang, X., Shum, H.: A bayesian mixture model for multi-view face alignment. In: Proceedings of Computer Vision and Pattern Recognition (2005)
11. Gu, L., Kanande, T.: 3d alignment of face in a single image. In: Computer Vision and Pattern Recognition (2006)
12. Zhang, Z., Liu, Z., Adler, D., Cohen, M.F., Hanson, E., Shan, Y.: Robust and rapid generation of animated faces from video images - a model-based modeling approach. International Jornal of Computer Vision (2004)
13. Liang, L., Wen, F., Xu, Y.Q., tang, X., Shum, H.Y.: Accurate face alignment using shape constrained markov network. In: Computer Vision and Pattern Recognition (2006)
14. Amit, Y., Kong, A.: Graphical templates for model registration. IEEE Trans. Pattern Anal. Mach. Intell. 18, 225–236 (1996)

15. Coughlan, J., Yuille, A., English, C., Snow, D.: Efficient optimization of a deformable template using dynamic programming. In: Computer Vision and Pattern Recognition, pp. 747–752 (1998)
16. Coughlan, J., Ferreira, S.: Finding deformable shapes using loopy belief propagation. In: Tistarelli, M., Bigun, J., Jain, A.K. (eds.) ECCV 2002. LNCS, vol. 2359, pp. 453–468. Springer, Heidelberg (2002)
17. Ahonen, T., Hadid, A., Pietikainen, M.: Face recognition with local binary patterns. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 469–481. Springer, Heidelberg (2004)
18. Cristinacce, D., Cootes, T.: Boosted regression active shape models. In: British Machine Vision Conference, pp. 880–889 (2007)
19. Zhou, Y., Gu, L., Zhang, H.: Bayesian tangent shape model: Estimating shape and pose parameters via bayesian inference. In: Computer Vision and Pattern Recognition, pp. 109–116 (2003)
20. Tipping, M.E., Bishop, C.M.: Mixtures of probabilistic principal component analysers. Neural Computation 11, 443–482 (1999)
21. Rowley, H., Baluja, S., Kanade, T.: Rotation invariant neural network-based face detection. In: Computer Vision and Pattern Recognition, pp. 38–44 (1998)
22. Gross, R., Matthews, I., Cohn, J., Baker, S.: Guide to the cmu multi-pie face database. Technical report, CMU RI (2002)
23. Martinez, A., Benavente, R.: The ar face database. CVC Technical Report 24 (1998)

Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs

Xiaowei Li, Changchang Wu, Christopher Zach,
Svetlana Lazebnik, and Jan-Michael Frahm

Dept. of Computer Science, University of North Carolina
Chapel Hill, NC 27599-3175
{xwli,ccwu,cmzach,lazebnik,jmf}@cs.unc.edu

Abstract. This paper presents an approach for modeling landmark sites such as the Statue of Liberty based on large-scale contaminated image collections gathered from the Internet. Our system combines 2D appearance and 3D geometric constraints to efficiently extract scene summaries, build 3D models, and recognize instances of the landmark in new test images. We start by clustering images using low-dimensional global "gist" descriptors. Next, we perform geometric verification to retain only the clusters whose images share a common 3D structure. Each valid cluster is then represented by a single iconic view, and geometric relationships between iconic views are captured by an *iconic scene graph*. In addition to serving as a compact scene summary, this graph is used to guide structure from motion to efficiently produce 3D models of the different aspects of the landmark. The set of iconic images is also used for recognition, i.e., determining whether new test images contain the landmark. Results on three data sets consisting of tens of thousands of images demonstrate the potential of the proposed approach.

1 Introduction

The recent explosion in consumer digital photography and the phenomenal growth of photo-sharing websites such as Flickr.com have created a high demand for computer vision techniques for creating effective visual models from large-scale Internet-based image collections. Given a large database of images downloaded using a keyword search, the challenge is to identify all photos that represent the concept of interest and to build a coherent visual model of the concept despite heavy contamination by images with wrongly associated tags.

Recent literature contains a number of approaches that address the problem of visual learning from Internet image collections for general object categories (see, e.g., [1,2,3]). These approaches are well-adapted to deal with label uncertainty and make effective use of statistical appearance-based modeling, but lack strong geometric constraints that are needed for modeling categories with a common rigid 3D structure, such as famous tourist sites and landmarks. For modeling and visualization of landmarks from Internet images, structure-from-motion methods methods have been proposed [4,5]. These methods employ powerful geometric

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 427–440, 2008.

constraints and produce compelling 3D reconstructions, but are currently not very scalable and not well suited to take advantage of more than a small subset of a large and noisy community photo collection.

This paper presents a hybrid approach that combines the strengths of 2D recognition and 3D reconstruction for representing landmarks based on images downloaded from Flickr.com using keyword searches. Our system proceeds in an incremental fashion, initially applying 2D appearance-based constraints to loosely group images, and progressively refining these groups with geometric constraints to select *iconic images* for a sparse visual summary of the scene. These images and the pairwise geometric relationships between them define an *iconic scene graph* that captures all the salient aspects of the landmark. The iconic scene graph is then used for efficient reconstruction of a 3D skeleton model which can also be extended to many more relevant images to a comprehensive "collective representation" of the scene. The process of registering new test images to the model also allows us to answer the recognition question, namely, whether the landmark of interest is visible in a new test image. In addition, the iconic scene graph can be used to organize the image collection into a hierarchical browsing system. Because our method prunes many spurious images using fast 2D constraints and applies computationally demanding geometric constraints to just a small subset of "promising" images, it is scalable to large photo collections.

2 Previous Work

This paper offers a comprehensive solution to the problems of dataset collection, 3D reconstruction, scene summarization, browsing and recognition for landmark images. Below, we discuss related recent work in these areas.

The problem of *dataset collection* refers to the following: starting with the heavily contaminated output of an Internet image search query, extract a high-precision subset of images that are actually relevant to the query. Existing approaches to this problem [1,2,3] consider general visual categories not necessarily related by rigid 3D structure. They use statistical models to combine different kinds of 2D image features (texture, color, keypoints), as well as text and tags. However, 2D features alone do not provide strong enough constraints when applied to landmark images. Given the amount of clutter, viewpoint change, and lighting variation typically present in consumer snapshots, as well as the unreliability of user-supplied tags, it is difficult to answer the question of whether a landmark is actually present in a given picture without bringing in structure-from-motion (SFM) constraints.

The *Photo Tourism* system of Snavely et al. [5] uses SFM constraints very effectively for modeling and visualization of landmarks. This system achieves high-quality reconstruction results with the help of exhaustive pairwise image matching and global bundle adjustment after inserting each new view. Unfortunately, this process becomes very computationally expensive for large data sets, and it is especially inefficient for heavily contaminated collections, most of whose images cannot be registered to each other. Accordingly, the input images used by Photo Tourism have either been acquired specifically for the task, or

downloaded and pre-filtered by hand. When faced with a large and heterogeneous dataset, the best this method can do is use brute force to reduce it to a small subset that gives a good reconstruction. For example, for the Notre Dame results reported in [5] 2,635 images of Notre Dame were used initially, and out of these, 597 images were successfully registered after about two weeks of processing.

More recently, several researchers have developed SFM methods that exploit the redundancy in community photo collections to make reconstruction more efficient. In particular, many landmark image collections consist of a small number of "hot spots" from which photos are taken. Ni et al. [6] have proposed an out-of-core bundle adjustment approach that takes advantage of this by locally optimizing the "hot spots" and then connecting the local solutions into a global one. In this paper, we follow a similar strategy of computing separate 3D reconstructions on connected sub-components of the scene, thus avoiding the need for frequent large-scale bundle adjustment. Snavely et al. [7] find *skeletal sets* of images from the collection whose reconstruction provides a good approximation to a reconstruction involving all the images. Similarly, our method is based on finding a small subset of *iconic images* that capture all the important aspects of the scene. However, unlike [6,7], we rely on 2D appearance similarity as a "proxy" or a rough approximation of the "true" multi-view relationship, and our goals are much broader: in addition to reconstruction, we are also interested in summarization, browsing, and recognition.

The problem of scene summarization for landmark image collections has been addressed by Simon et al. [8], who cluster images based on the output of exhaustive pairwise feature matching. While this solution is effective, it is perhaps too "strong" for the problem, as in many cases, a good subset of representative or "iconic" images can be obtained for a scene using much simpler 2D techniques [9]. This is the philosophy followed in our work: instead of treating scene summarization as a by-product of SFM, we treat it as a first step toward efficiently computing the scene structure.

Another problem relevant to our work is that of retrieval: given a query image, find all images containing the same landmark in some target database [10,11]. In this paper, we use retrieval techniques such as fast feature-based indexing and geometric verification with RANSAC to establish geometric relationships between different iconic images and to register a new test image to the iconics for the purpose of recognition.

3 The Approach

In this section, we present the components of our implemented system. Figure 1 gives a high-level summary of these components, and Figure 2 illustrates them with results on the Statue of Liberty dataset.

3.1 Initial Clustering

Our goal is to compute a representation of a landmark site by identifying a set of *canonical* or *iconic* views corresponding to dominant scene aspects. Recently,

1. **Initial clustering** (Section 3.1): Use "gist" descriptors to cluster the collection into groups roughly corresponding to similar viewpoints and scene conditions.
2. **Geometric verification and iconic image selection** (Section 3.2): The goal of this stage is to filter out the clusters whose images do not share a common 3D structure. This is done by pairwise epipolar geometry estimation among a few representative images selected from each cluster. The image that gathers the most inliers to the other representative images in its cluster is selected as the *iconic image* for that cluster.
3. **Construction of iconic scene graph** (Section 3.3): Perform pairwise epipolar geometry estimation among the iconic images and create an *iconic scene graph* by connecting pairs of iconics that are related by a fundamental matrix or a homography. Edges are weighted by the number of inliers to the transformation.
4. **Tag-based filtering** (Section 3.4): Use tag information to reject isolated nodes of the iconic scene graph that are semantically irrelevant to the landmark.
5. **3D reconstruction** (Section 3.5): First, partition the iconic scene graph into several tightly connected components and compute structure from motion separately on each component. Within each component, use a maximum spanning tree to determine the order of registering images to the model. At the end, merge component models along cut edges.
6. **Recognition** (Section 4.2): Given a new test image, determine whether it contains an instance of the landmark. This can be done by efficiently registering the image to the iconics using appearance-based scores and geometric verification.

Fig. 1. Summary of the major steps of our system

Simon et al. [8] have defined iconic views as representatives of dense clusters of similar viewpoints. To find these clusters, Simon et al. take as input a feature-view matrix (a matrix that says which 3D features are present in which views) and define similarity of any two views in terms of the number of 3D features they have in common. By constrast, we adopt a perceptual or image-based perspective on iconic view selection: if there are many images in the dataset that share a very similar viewpoint in 3D, then at least some of them will have a very similar image appearance in 2D, and can be matched efficiently using a low-dimensional global description of their pixel patterns.

The global descriptor we use is *gist* [12], which was found to be effective for grouping images by perceptual similarity [13]. We cluster the gist descriptors of all our input images using k-means with $k = 1200$. Since at the next stage, we will select at most a single iconic image to represent each cluster, we initially want to produce an over-clustering to give us a large and somewhat redundant set of candidate iconics. In particular, we can expect images with very similar viewpoints to end up in different gist clusters because of clutter (i.e., people in front of the camera), differences in lighting, or camera zoom. This does not cause a problem for our approach, because the graph construction step of Section 3.3 will be able to restore links between different clusters that have sufficient viewpoint similarity.

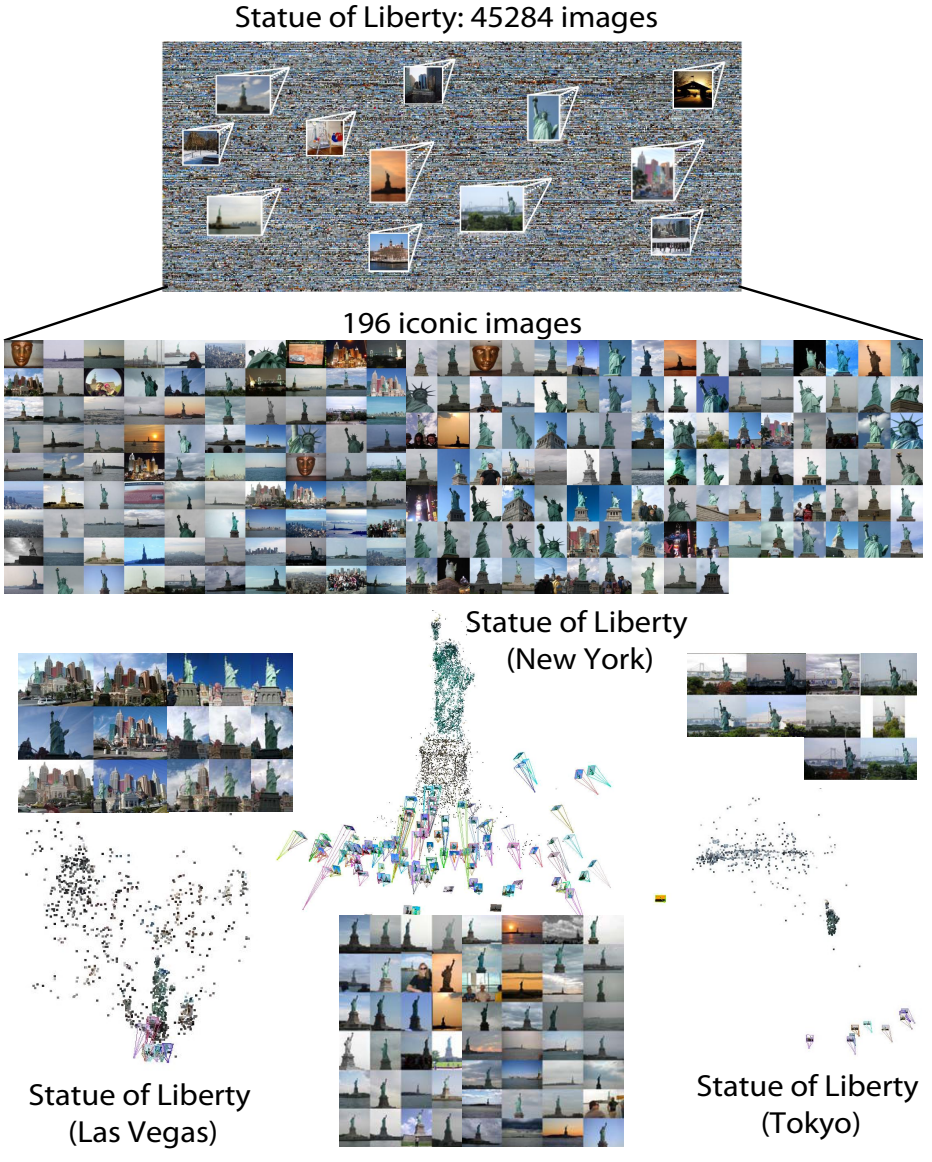


Fig. 2. A snapshot of the operation of our system for the Statue of Liberty. The initial dataset of 45284 images (of which about 40% are unrelated to the landmark) gets reduced to a set of 196 iconics by 2D appearance-based clustering followed by geometric verification of top cluster representatives. The iconics are nodes in a scene graph consisting of multiple connected components, each of which gives rise to a 3D reconstruction. The largest reconstructed component corresponds to the Statue of Liberty in New York, while two of the smaller ones correspond to copies of the statue in Tokyo and Las Vegas. These components are found completely automatically. A video of the models can also be found at `http://www.cs.unc.edu/iconic-scene-graphs`.

In our experiments, we have found that the largest gist clusters tend to be the cleanest ones. Therefore, for the initial stage, we use cluster size to produce a first, coarse ranking of images. As shown in the quantitative evaluation in Figure 6(a) (Stage 1), the first few gist clusters have a surprisingly high precision, though it deteriorates rapidly for subsequent clusters.

3.2 Geometric Verification and Iconic Image Selection

The next step is to perform verification of each gist cluster to confirm that its images share a common 3D structure. To do this efficiently, we select a small number of the most representative images from each cluster and attempt to estimate the two-view geometry of every pair. The representatives are given by n images (in our current implementation, $n = 8$) whose gist descriptors are closest to the cluster mean. Note that gist clusters that have fewer than n images are rejected before this step.

For fitting a geometric transformation to these matches, we extract SIFT features [14] and use QDEGSAC [15], which is a robust procedure that returns an estimate for a fundamental matrix or a homography, depending on the scene structure. The image that gathers the largest total number of inliers to the other $n-1$ representatives from its cluster is declared the *iconic image* of that cluster. If any of the remaining representatives are not consistent with the iconic, we remove them and attempt to replace them with other images from the same cluster (within clusters, images are ranked in order of increasing distance from the center). If at the end of this process we are not able to find $n-1$ other consistent images, the cluster is rejected.

The inlier score of each iconic can be used as a new measure of the quality of each cluster. Precision/recall curves in Figure 6(a) (Stage 2) demonstrate that this ranking does a better job than gist alone in separating the relevant images from the irrelevant ones. However, there is an undesirable effect of a few geometrically consistent, but semantically irrelevant clusters getting very high scores at this stage, which hurts precision early on. Such clusters typically result from near-duplicate images coming from the same user's photo album. As described in the next section, we will be able to reject many such clusters using inter-cluster matching and filtering based on tags.

Ranking of clusters based on the top representatives does not penalize clusters that have a few geometrically consistent images, but are very low-precision otherwise. Once the iconic images for every cluster are selected, we can perform geometric verification of every remaining image by matching it to the iconic of its cluster and ranking it individually by the number of inliers it has with respect to its iconic. As shown in Figure 6(a) (Stage 3), this individual ranking improves precision considerably.

3.3 Construction of Iconic Scene Graph

Next, we need to establish links between the iconic images selected in the previous step. Since we have hundreds of iconic images even following rejection of

geometrically inconsistent clusters, exhaustive pairwise matching of all iconics is still rather inefficient. To match different iconic images, we need to account for larger viewpoint and appearance changes than in the initial clustering, so keypoint-based methods are more appropriate for this stage. We use the vocabulary tree method of Nister and Stewenius [16] as a fast indexing scheme to obtain promising candidates for pairwise geometric verification. We train a vocabulary tree with five levels and a branching factor of 10 using a set of thousands of frames taken from a video sequence of urban data, and populate it with SIFT features extracted from our iconics. We then use each iconic as a query image and perform geometric verification with top 20 other iconics returned by the vocabulary tree. Pairs of iconics that match with more than 18 inliers are then connected by an edge whose weight is given by the inlier score. This results in an undirected, weighted *iconic scene graph* whose nodes coresond to iconic views and edges correspond to two-view transformations (homographies or fundamental matrices) relating the iconics.

Next, we would like to identify a small number of strongly connected components in the iconic scene graph. This serves two purposes. The first is to group together iconics that are close in terms of viewpoint but did not initially fall into the same gist cluster. The second is to obtain smaller subsets of images on which structure from motion can be performed more efficiently. To partition the graph, we use normalized cuts [17], which requires us to specify as input the desired number of components. This parameter choice is not critical, since any oversegmentation of the graph will be addressed by the component merging step discussed in the next section. We have found that specifying a target number of 20 to 30 components produces acceptable results for all our datasets.

Every (disjoint) component typically represents a distinctive aspect of the landmark, and we can select a single representative iconic for each component (i.e., the iconic with the highest sum of edge weights) to form a compact scene summary. Moreover, the components of the iconic scene graph and the iconic clusters induce a hierarchical structure on the dataset that can be used for browsing, as shown in Figure 3.

3.4 Tag-Based Filtering

The iconic scene graph tends to have many isolated nodes, corresponding to iconic views for which we could not find a geometric relationship with any other view. These nodes are excluded from the graph partitioning process described above. They may either be aspects of the scene that are significantly different from others, e.g., the interior of the Notre Dame cathedral in Paris, or geometrically consistent, but semantically irrelevant clusters, e.g., pictures of a Notre Dame cathedral in a different city. Since constraints on appearance and geometry are not sufficient to establish the relationship of such clusters to the scene, to refine our dataset further we need to use additional information, in particular, the tags associated with the images on Flickr.

Even though Flickr tags in general tend to be quite unreliable, we have observed that among the isolated clusters that have already been pre-filtered by

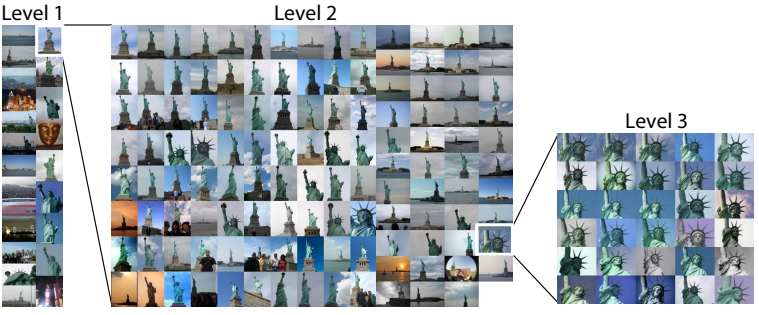


Fig. 3. Hierarchical organization of the dataset for browsing. Level 1: components of the iconic scene graph. Level 2: Each component can be expanded to show all the iconic images associated with it. Level 3: each iconic can be expanded to show the images associated with its gist cluster. Our three datasets may be browsed online at `http://www.cs.unc.edu/iconic-scene-graphs`.

appearance and geometry constraints, there are quite a few whose tags are clearly unrelated to the landmark. This suggests that, provided we have a good idea of the distribution of relevant tags, a very simple score should be sufficient to identify the "bad" clusters. Fortunately, during the previous modeling stages, we have already verified hundreds of images without resorting to tags, so we can now use these images to acquire the desired distribution. In the implementation, we take iconic images that have at least two edges in the scene graph (empirically, these are almost certain to contain the landmark), and use them to create a "master list" of relevant tags. To have a more complete list, we also incorporate tags from the top cluster images registered to these iconics. The tags in the list are ranked in decreasing order of frequency, and isolated iconic images are scored based on the median rank of their tags (tags that don't occur in the master list at all are assigned an arbitrary high number). Clusters with "obviously" unrelated tags get a high median rank and can be removed to increase precision, as shown by the "Stage 4" curves in Figure 6(a).

3.5 3D Reconstruction

As a first step, the 3D structure for every component produced by normalized cuts is computed separately. Starting with a good initial image pair, we incrementally add more views to the reconstruction by perspective pose estimation. There are two criteria for selecting a good initial pair. First, in order to operate in metric instead of projective space, the views in question require reasonable estimates for the focal lengths, which can be obtained from EXIF data. When EXIF data is not available, we transfer the focal length estimate from similar views in the same cluster. Second, the number of inlier correspondences should be as large as possible, taking into account the 3D point triangulation certainty (i.e. the shape of the covariance matrices) [18]. Once an initial pair of images is found, their relative pose is determined by the five-point method [19]. The

Table 1. Summary statistics of our datasets and 3D models. The first five columns list dataset sizes and numbers of labeled images. The next two columns give details of our computed 3D models: the number of distinct models after merging and the total number of registered views (these include both iconic and non-iconic images). The last two columns refer to just the single largest 3D model. They list the number of registered views and the number of 3D points visible in at least three views.

	Modeling			Testing		All 3D models		Largest 3D model	
Dataset	Unlabeled	Pos.	Neg.	Pos.	Neg.	#Models	#Views	#Views	#Pts
Notre Dame	9760	545	535	541	503	8	580	337	30802
Statue of Liberty	42983	1369	932	646	446	6	1068	871	18675
San Marco	38332	2094	3131	379	715	4	1213	749	39307

remaining views of the current component are added using perspective pose estimation from 2D-3D correspondences. The order of insertion is determined from the edges of the underlying maximum spanning tree computed for the weighted graph component. Hence, views having more correspondences with the current reconstruction are added first. The resulting 3D structure and the camera parameters are optimized by non-linear sparse bundle adjustment [20].

The above reconstruction process applied to individual graph components produces small individual models representing single aspects of the landmark. A robust estimation of the 3D similarity transform is used to align and merge suitable components and their respective reconstructions. This merging process restores the connectivity of the original scene graph, and results in a single "skeleton" 3D model for each original connected component. The last step is to augment these models by incorporating additional non-iconic images from clusters. Each image we consider for adding has already been successfully registered with the iconic of its cluster, as described in Section 3.2. Since the features from the iconic have already been incorporated into the skeleton 3D model, the registration between the image and the iconic gives us a number of 2D/3D matches for that image. To obtain additional 2D/3D matches, we attempt to register this new image to two additional iconics that are connected to its original iconic by the highest-weight edges. All these matches are then used to estimate the pose of the new image. At the end, bundle adjustment is applied to refine the model.

4 Experimental Results

4.1 Data Collection and Model Construction

We have tested our system on three datasets: the Notre Dame cathedral in Paris, the Statue of Liberty in New York, and Piazza San Marco in Venice. The datasets were automatically downloaded from Flickr.com using keyword searches. We randomly split each dataset into a "modeling" part, and a much smaller independent "testing" part. Because the modeling datasets contain tens of thousands of images, we have chosen to label only a small randomly selected fraction of them. These ground-truth labels are needed only to measure recall and precision for the different stages of refinement, since our modeling approach

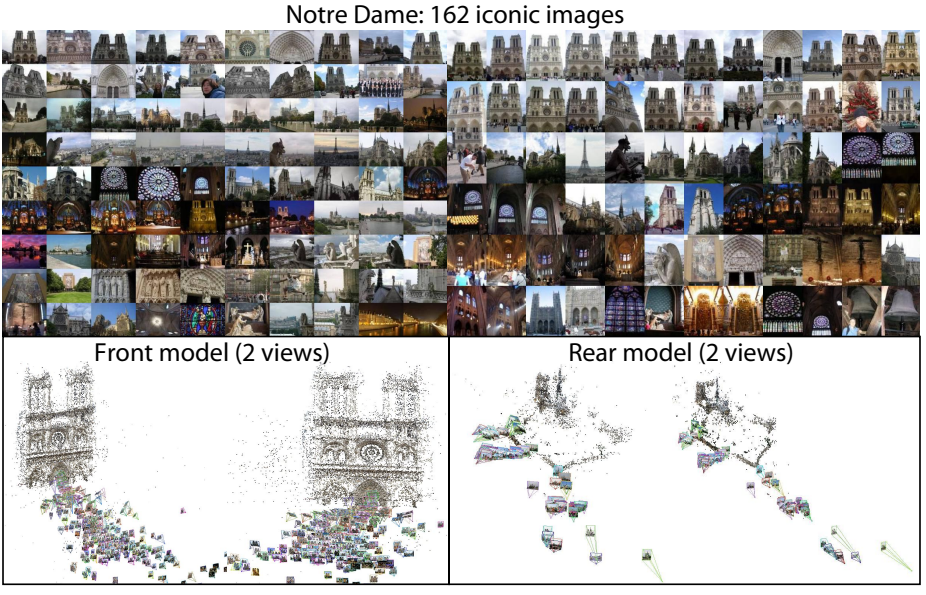


Fig. 4. Notre Dame results. Top: iconic images. Note that there are a few spurious iconics corresponding to Notre Dame cathedrals in Indiana and Montreal (these were not removed by the tag filtering step), as well as the Eiffel Tower seen from the top of Notre Dame. Bottom: two of the reconstructed scene components, corresponding to the front and back of the cathedral.

is completely unsupervised. The smaller test sets are completely labeled. Our labeling is very basic, merely recording whether the landmark is present in the image or not, without evaluating the "quality" or "typicality" of a given view. For example, interior views of Notre Dame are labeled as positive, even though they are relatively few in number and cannot be registered to the exterior views. Table 1 gives a breakdown of the numbers of labeled and unlabeled images in our datasets. The proportions of negative images (40% to 60%) give a good idea of the initial amount of contamination.

Figure 6(a) shows recall/precision curves for the modeling process on the three datasets. We can see that the four refinement stages (gist clustering, geometric verification of clusters, verification of individual images w.r.t. their cluster centers, and tag-based rejection) progressively increase the precision of the images selected as part of the model, even though recall decreases following the rejection decisions made after every stage.

Figures 2, 4 and 5 show the reconstructed 3D models for our three datasets (see `http://www.cs.unc.edu/iconic-scene-graphs` for videos of the models). As described in Section 3.5, reconstruction is first performed on separate components of the iconic scene graph, followed by merging of models with sufficiently overlapping scene components. Successful merging requires images that link the component models. For San Marco, the merging of the models corresponding to

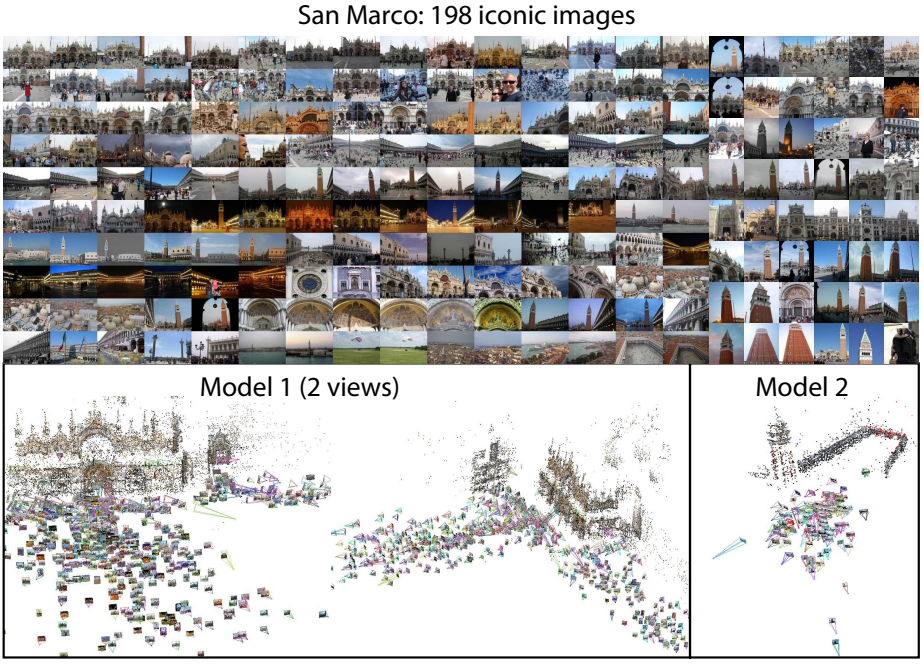


Fig. 5. San Marco results. Top: iconic images. Bottom: two of the reconstructed scene components, corresponding to the front and the back of the square.

the front and back of the square was not successful because the iconic images did not provide a sufficient coverage of the middle region. For a similar reason, it was not possible to merge the front and the back of Notre Dame. To an extent, this merging problem is endemic to community photo collections, as people tend to take snapshots of famous landmarks from a small number of particularly characteristic or accessible "hot spots," while the areas in between remain sparsely covered. Our clustering approach may in some cases exacerbate this problem by discarding the less common images that fail to produce sufficiently large clusters. Despite the difficulty of merging, our models successfully incorporate a significant number of images, as shown in Table 1. While our models currently do not exceed in size those produced by the Photo Tourism system [5], we are able to process an order of magnitude more images with just a fraction of the computational power, i.e., hours on a single commodity PC, instead of weeks on a high-performance cluster.

4.2 Testing and Recognition

Given a new image that was not in our initial collection, we want to find out whether it contains the landmark of interest. A straightforward of doing this is by retrieving the iconic image that gets the highest matching score with the

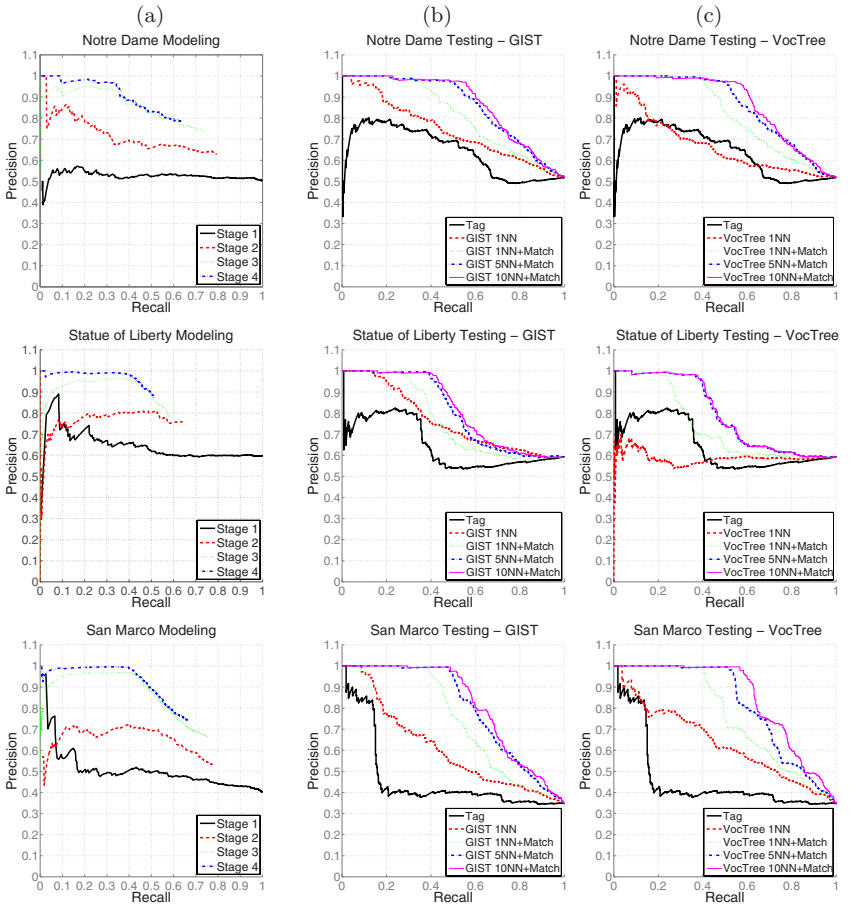


Fig. 6. Recall/precision curves for (a) modeling; (b) testing with the gist descriptors; and (c) testing with the vocabulary tree. For modeling, the four stages are as follows. **Stage 1:** Clustering using gist and ranking each image by the size of its gist cluster (Section 3.1). **Stage 2:** Geometric verification of iconics and ranking each image by the inlier number of its iconic (Section 3.2). The recall is lower because inconsistent clusters are rejected. **Stage 3:** Registering each image to its iconic and ranking the image by the number of inliers of the two-view transformation to the iconic. Unlike in the first two stages, images are no longer arranged by cluster, but ranked individually by this score. The recall is lower because images with not enough inliers to estimate a two-view transformation are rejected. **Stage 4:** Tag information is used to retain only the top 30 isolated clusters (Section 3.4). The score is the same as in stage 3, except that images belonging to the rejected clusters are removed. Note the increase in precision in the first few retrieved images. For testing, the different retrieval strategies are as follows. **GIST 1NN** (resp. **VocTree 1NN**): retrieval of the single nearest iconic using the gist descriptor (resp. vocabulary tree); **GIST kNN+Match** (resp. **VocTree kNN+Match**): retrieval of k nearest exemplars using gist (resp. vocabulary tree) followed by geometric verification; **Tag:** tag-based ranking (see Section 3.4).

test image (according to a given retrieval scheme) and making the yes/no decision by setting a threshold on the retrieval score. We can evaluate performance quantitatively by plotting a recall/precision curve of the test images ordered from highest to lowest score. Figure 6 (b) and (c) shows the results for several retrieval strategies. The simplest strategy is to compare the test image to the iconics using either gist descriptors (in which case the score would be inversely proportional to the distance) or a bag-of-features representation using the vocabulary tree (which returns a tf/idf score [16]). For improved performance, we can take top k "candidate" iconics retrieved with either gist or vocabulary tree, and perform geometric verification with each candidate as described in Section 3.2. In this case, the score for each candidate is the number of inliers to a two-view transformation (homography or fundamental matrix) between it and the test image, and only the top candidate is retained.

Interestingly, for the Statue of Liberty, the performance of the vocabulary tree without geometric verification is almost disastrous. This is due to the relative lack of texture in many Statue of Liberty images, which gives too few local features for bag-of-words matching to work reliably. But in most other cases, gist and vocabulary tree have comparable performance. Not surprisingly, for both kinds of image description, geometric verification significantly improves accuracy, as does retrieving more candidates for verification. For comparison, we also include a recall/precision curve for scoring test images based on their tag relevance (see Section 3.4). By itself, this scoring scheme is quite unreliable.

5 Discussion

We have presented a hybrid approach combining 2D appearance and 3D geometry to efficiently model and recognize complex real-world scenes captured by thousands of amateur photos. To our knowledge, our system is the first integrated solution to the problems of dataset collection, scene summarization, browsing, 3D reconstruction, and recognition for landmark images. At the heart of our approach is the *iconic scene graph*, which captures the major aspects of the landmark and the geometric connections between them. The structure of this graph is used, among other things, to create a three-level browsing hierarchy and to enable scalable computation of structure from motion.

In the future, one of our main goals is to improve the recall of our modeling process. This can be done by making use of fast and accurate retrieval techniques such as [10,11] to re-incorporate images that were discarded during the iconic image selection stage. A similar strategy could also be helpful for discovering "missing links" for merging 3D models of different components. In addition, we plan to create 3D models that incorporate a much larger number of images. This will require a memory-efficient streaming approach for registering new images, as well as out-of-core bundle adjustment using iconic scene graph components.

References

1. Fergus, R., Perona, P., Zisserman, A.: A visual category filter for Google images. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 242–256. Springer, Heidelberg (2004)
2. Berg, T., Forsyth, D.: Animals on the web. In: CVPR (2006)
3. Schroff, F., Criminisi, A., Zisserman, A.: Harvesting image databases from the web. In: ICCV (2007)
4. Goesele, M., Snavely, N., Curless, B., Hoppe, H., Seitz, S.M.: Multi-view stereo for community photo collections. In: ICCV (2007)
5. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3d. In: SIGGRAPH, pp. 835–846 (2006)
6. Ni, K., Steedly, D., Dellaert, F.: Out-of-core bundle adjustment for large-scale 3d reconstruction. In: ICCV (2007)
7. Snavely, N., Seitz, S.M., Szeliski, R.: Skeletal sets for efficient structure from motion. In: CVPR (2008)
8. Simon, I., Snavely, N., Seitz, S.M.: Scene summarization for online image collections. In: ICCV (2007)
9. Berg, T.L., Forsyth, D.: Automatic ranking of iconic images. Technical report, University of California, Berkeley (2007)
10. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: ICCV (2007)
11. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: CVPR (2008)
12. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. IJCV 42(3), 145–175 (2001)
13. Hays, J., Efros, A.A.: Scene completion using millions of photographs. In: SIGGRAPH (2007)
14. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
15. Frahm, J.M., Pollefeys, M.: RANSAC for (quasi-)degenerate data (QDEGSAC). In: CVPR, vol. 1, pp. 453–460 (2006)
16. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR (2006)
17. Shi, J., Malik, J.: Normalized cuts and image segmentation. PAMI 22, 888–905 (2000)
18. Beder, C., Steffen, R.: Determining an initial image pair for fixing the scale of a 3d reconstruction from an image sequence. In: Proc. DAGM, pp. 657–666 (2006)
19. Nistér, D.: An efficient solution to the five-point relative pose problem. PAMI 26, 756–770 (2004)
20. Lourakis, M., Argyros, A.: The design and implementation of a generic sparse bundle adjustment software package based on the Levenberg-Marquardt algorithm. Technical Report 340, Institute of Computer Science - FORTH (2004)

VideoCut: Removing Irrelevant Frames by Discovering the Object of Interest

David Liu[1], Gang Hua[2], and Tsuhan Chen[1]

[1] Dept. of ECE, Carnegie Mellon University
[2] Microsoft Live Labs
dliu@cmu.edu, ganghua@microsoft.com, tsuhan@cmu.edu

Abstract. We propose a novel method for removing irrelevant frames from a video given user-provided frame-level labeling for a very small number of frames. We first hypothesize a number of candidate areas which possibly contain the object of interest, and then figure out which area(s) truly contain the object of interest. Our method enjoys several favorable properties. First, compared to approaches where a single descriptor is used to describe a whole frame, each area's feature descriptor has the chance of genuinely describing the object of interest, hence it is less affected by background clutter. Second, by considering the temporal continuity of a video instead of treating the frames as independent, we can hypothesize the location of the candidate areas more accurately. Third, by infusing prior knowledge into the topic-motion model, we can precisely follow the trajectory of the object of interest. This allows us to largely reduce the number of candidate areas and hence reduce the chance of overfitting the data during learning. We demonstrate the effectiveness of the method by comparing it to several other semi-supervised learning approaches on challenging video clips.

1 Introduction

The endless streams of videos on the Internet often contain irrelevant data. Our goal is to cut video clips shorter and retain the frames that are relevant to the user input. We assume the user has an "*object of interest*" (OOI) in mind, which can, for example, be a car, a book, or the scene of a forest. The system will infer which frames contain the OOI. This application can be used, e.g., for shortening surveillance videos or TV programs.

We consider the case where the system is provided with very limited information. Specifically, the user will label at least one frame as relevant and another frame as irrelevant. These labels are at the frame-level instead of at the pixel-level. Although pixel-level labeling (such as using a bounding box or segmentation mask to specify the location of the OOI) can provide more information, we intend to explore the possibility of letting the user provide coarser and less tedious labeling.

We formulate the task as a self-training multiple instance learning problem. For each frame, we postulate a number of candidate areas, and use a multiple

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 441–453, 2008.

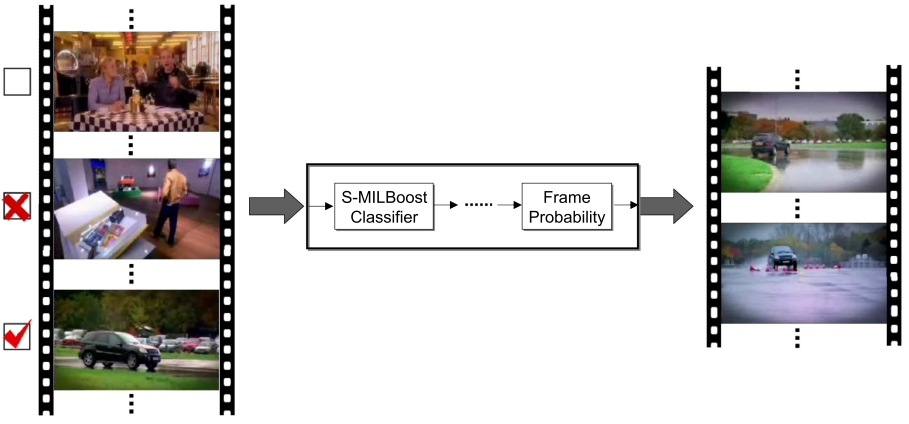


Fig. 1. Frames are unlabeled (top left), labeled as irrelevant (middle left) or relevant (bottom left). The system will find out what the object of interest is (in this case, the black vehicle) and remove frames that don't contain the vehicle.

instance learning algorithm to simultaneously find out whether the OOI exists in the frame, and if it does, where it is located. The reason that we go one step beyond our goal (that is, trying to locate the OOI) is because we are able to exploit the temporal smoothness property of video objects to consolidate their locations. That is to say, objects tend to move in a continuous manner from frame to frame.

We use sporadically labeled frames to train a multiple instance learning algorithm called MILBoost [22]. It was originally applied to a face detection problem. In their work, images are manually labeled by drawing a rectangle around the head of a person. In our system, we only have frame-level labels, i.e., no rectangles are available.

Our semi-supervised framework can be distinguished from prior work in several aspects. Our work does not require pixel-level labeled data. In[17], learning requires both pixel-level labeled data and frame-level labeled data. An object detector is initially trained on the pixel-level labeled data, and the learned model is used to estimate labels for the frame-level labeled data. As illustrated in Fig. 1, we "**discover**" the OOI since no bounding box is given, which also distinguishes our work with the video object retrieval work in [20][19], where the OOI is explicitly labeled at the pixel-level.

Image retrieval systems often allow users to provide positive and negative feedback, hence the task of image retrieval can also be cast under the self-training [14] or multiple instance learning [22] framework. Nonetheless, our system exploits temporal information of videos in a novel way, which distinguishes itself from the image retrieval literature. In [16], activities in a video are condensed into a shorter period by simultaneously showing multiple activities. It does not intend to **discover** the frames that contain the user-desired OOI from limited user input.

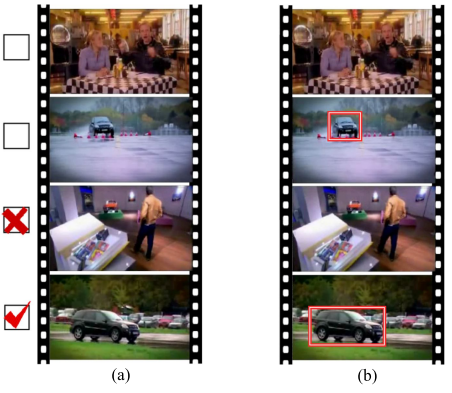


Fig. 2. (a) Labeling at the frame level assumed in this work. Frames can be unlabeled, or labeled as positive or negative. (b) The bounding box type of labeling provides more explicit information regarding the object of interest, but is also more tedious in the labeling process.

Our method is based on the bag-of-words representation, which is part-based. Different than other part-based methods such as the one-shot learning framework [7], we leverage motion consistency to improve recognition, while the one-shot learning framework did not utilize that. We leverage the unsupervised topic-motion model in [11] and extend it to a semi-supervised setting by incorporating additional prior models during learning. The problem solved, the application targeted, as well as the fundamental approach adopted in our paper, are significantly different from [11].

Our contribution can hence be summarized as follows: **1)** A novel application that summarizes videos based on the implicitly specified OOI. **2)** A novel system that uses weakly labeled data for object discovery in video. **3)** A novel method that takes advantage of the temporal smoothness property during semi-supervised learning.

The paper is organized as follows. In section 2 we define the type of user labeling information that is available to the system. In section 3 we introduce a baseline method, where features at the frame-level are used for semi-supervised learning. In section 4 we explain in detail the proposed method. In section 5 we will compare the proposed method with the baseline method and several variants of the proposed method. Finally, we conclude in section 6.

2 Frame-Level Labels

The amount of user label information as well as its format has a major impact on system design. The amount of user label information can range from all frames being labeled to none. For those frames being labeled, the labeling can be as detailed as providing bounding boxes for each frame (which we call pixel-level labeling), or as coarse as "this frame does (or does not) contain the OOI" (which we call frame-level labeling).

In this paper, we consider the more challenging task of having as input only frame-level labeling; see Fig. 2 for a comparison. This kind of 'weak labeling' is very different from traditional object detection; see for example [18], where the characteristics of the OOI are learned from plenty of pixel-level labeled data. This is also different from the recent video retrieval work in [20][19]. Traditional object detection not only involves a lot of human labor for labeling the images by putting bounding boxes on the OOI, but also has the difficulty of scaling to multiple categories of objects. Since the OOI in a sequence can be of any category, it is very difficult to train a comprehensive object detector that covers all types of objects.

3 Semi-supervised Learning at Frame-Level

Our first attempt to achieve the goal of VideoCut is to use semi-supervised learning at the frame-level. Each frame is represented as a histogram of *visual words*, or *textons* [9]. To generate visual words, we use the Maximally Stable Extremal Regions (MSER) operator [8] to find salient patches [1]. MSERs are the parts of an image where local contrast is high. Other operators could also be used; see [2] for a collection. Features are extracted from these MSERs by Scale Invariant Feature Transform (SIFT) [12]. In this work we extract MSERs and SIFT descriptors from grayscale images. Patches and features extracted from color images [21] can also be used instead. The SIFT features from a video are vector quantized using K-Means Clustering. The resulting $J = 50$ cluster centers form the dictionary of visual words, $\{w_1, ..., w_J\}$. Each MSER can then be represented by its closest visual word.

The histograms of the labeled frames along with their labels are fed to the system to train a classifier. The classifier is then applied to the unlabeled frames. Frames with high confidence scores are assigned pseudo-labels. The pseudo-labeled data is combined with the original labeled data and the classifier is trained again. The classifier we use is Discrete AdaBoost [4]. We will use this method as a baseline method in the experiments. This kind of self-training [14] procedure has been used extensively in different domains [10][17] and achieved top results in the NIPS competition [4].

4 Semi-supervised Learning at Sub-frame Level

There are two issues with the frame-level learning framework in Sec. 3.

1. The OOI can be small and the visual words from the whole frame are usually dominated by background clutter. Hence the full-frame histogram representation is not a truthful representation of the OOI.

[1] The word 'region' should not be confused with the 'candidate areas' to be introduced later. Each candidate area contains a set of MSER patches.

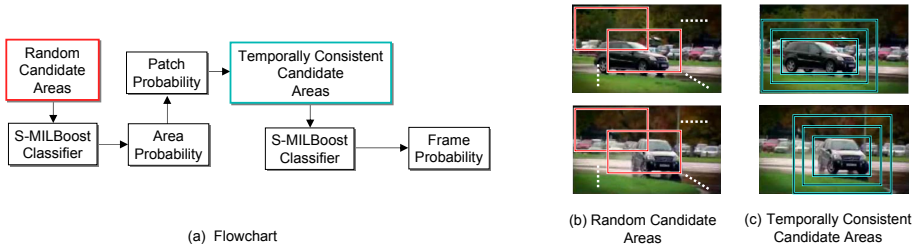


Fig. 3. Semi-supervised learning at sub-frame level using temporally consistent candidate areas

2. Objects in video often follow a smooth trajectory, which we call the *temporal smoothness property*. With frame-level learning, the temporal smoothness property cannot be readily exploited.

We address these issues by learning at a sub-frame level. Fig. 3(a) shows the proposed system flowchart. In each frame, we propose a number of *Random Candidate Areas* that potentially contain the OOI (illustrated in Fig. 3(b)). This will be detailed in section 4.1. The candidate areas are passed to a self-training version of MILBoost (S-MILBoost) and assigned an *Area Probability*, a score that tells us how likely this candidate area truly belongs to the OOI. This will be detailed in section 4.2. After each candidate area receives a score, we assign each image patch (MSER) a *Patch Probability*, which is defined as the largest *Area Probability* among the candidate areas that cover that image patch. Given the *Patch Probability*, in section 4.3 we will explain how to obtain the *Temporally Consistent Candidate Areas*. Basically, this is achieved by fitting a model which simultaneously *discovers* the OOI and *tracks* it across frames. The *Temporally Consistent Candidate Areas* are illustrated in Fig. 3(c); using them, we train S-MILBoost once again. As we will show in the experiments, this new S-MILBoost classifier will be more reliable than the previous one trained with the *Random Candidate Areas*. Finally, the S-MILBoost classifier gives us the *Frame Probability*, which tells us how likely each frame contains the OOI. Using the *Frame Probability*, we can determine the irrelevant frames and perform VideoCut.

Notice how the two issues mentioned earlier are resolved by using this proposed flowchart. First, the candidate areas are smaller than the whole frame and hence include less background clutter, which address the first issue mentioned above. Second, the candidate areas in one frame can be temporally correlated with the candidate areas in the next frame by performing 'weak' object tracking (illustrated in Fig. 3(c)), which addresses the second issue. We emphasize that this 'weak' tracking is different from traditional object tracking, as we will explain later.

In the experiments section we will compare our proposed flowchart with some other methods, which replace or omit some parts of the modules in Fig. 3(a). In the following subsections we will explain the details and merits of each component in Fig. 3(a).

4.1 Random Candidate Areas

Since the user labeling does not tell us where the OOI is located (neither in the labeled nor in the unlabeled frames), we need to set the candidate areas based on prior knowledge, if any. At the beginning, we use candidate areas with fixed size and uniform spacing and call them the random candidate areas. Each candidate area is represented as a histogram of visual words, as shown in Fig. 4. After we have a rough guess (using the techniques in the next two subsections), we will refine the candidate areas by placing them more densely around the estimated location of the OOI. We call these later candidate areas as temporally consistent candidate areas. See Fig. 3(b)(c) for illustrations.

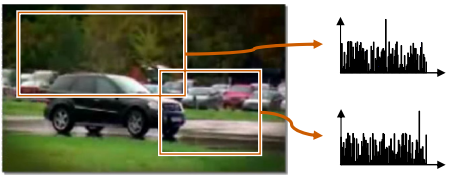


Fig. 4. Candidate areas, each represented by a histogram over visual words. In the experiments, we use a variety of different densities and spacings of candidate areas.

4.2 Self-training MILBoost

Using a similar self-training procedure as in Sec. 3, we first use the labeled frames to train a multiple instance learning [22] classifier. As a result, each candidate area of the labeled frames is assigned an area probability, which is the probability that an area contains the OOI. The classifier is then self-trained with the unlabeled frames and pseudo-labels included. As a result, the area probabilities of candidate areas in unlabeled frames are obtained as well.

Different than in Sec. 3, we have multiple histograms per frame, instead of a single one, therefore we use a multiple instance learning classifier, MILBoost [22]. First let us define some notations. We denote the histogram over visual words of a candidate area as $x_{k,a}$, where k indices over frames and a indices over the candidate areas inside frame k. Let $t_k \in \{0, 1\}$ denote the label or pseudo-label of frame k. Each frame has a *frame probability* p_k, and each candidate area has an *area probability* $p_{k,a}$. The *frame probability* is the probability that a frame contains the OOI, and the *area probability* is the probability that the area contains the OOI. Since a frame is labeled as positive as long as it contains the OOI, it is natural to model the relationship between p_k and $p_{k,a}$ using the Noisy-OR model [15], $p_k = 1 - \prod_{a \in k}(1 - p_{k,a})$. The likelihood is given by $L(C) = \prod_k {p_k}^{t_k}(1 - p_k)^{(1-t_k)}$.

As implied by its name, MILBoost produces a strong classifer $C(x_{k,a})$ in the form of a weighted sum of weak classifiers: $C(x_{k,a}) = \sum_u \lambda_u c_u(x_{k,a})$, $c_u(x_{k,a}) \in \{-1, +1\}$. The strong classifier score $C(x_{k,a})$ translates into the area probability, $p_{k,a}$, by the logistic sigmoid function $p_{k,a} = 1/(1 + exp(-C(x_{k,a})))$. Using the AnyBoost [13] method, the boosting weight $\varpi_{k,a}$ of each candidate area is the derivative of the log-likelihood, easily to be shown as $\frac{t_k - p_k}{p_k} p_{k,a}$. In round u of boosting, one first solves the optimization problem $c_u(.) = \arg\max_{c'(.)}$

$\sum_{k,a} c'(x_{k,a})\varpi_{k,a}$. A line search is then performed to seek for the optimal parameter λ_u, i.e., $\lambda_u = \arg\max_\lambda L(C + \lambda c_u)$.

In summary, S-MILBoost produces a classifier that assigns each frame a frame probability, and each candidate area an area probability. Notice that the S-MILBoost classifier is always used in a learning mode, during which the area and frame probabilities are estimated.

4.3 Temporally Consistent Candidate Areas

The accuracy of the frame probabilities depends heavily on the placing of the candidate areas; as an extreme example, if the OOI appears in a frame but none of the candidate areas cover it, then there would be no chance we could have correctly estimated the frame probability. This suggests a refinement of the placing scheme of candidate areas based on extra information. Notice that, we haven't yet exploited the temporal smoothness property of videos.

We would like to use the temporal smoothness property to refine the placing of the candidate areas. The temporal smoothness property is typically exploited through tracking the object. However, tracking requires manual initialization of the object location and size, information which is not available to us.

The topic-motion model [11] simultaneously estimates the appearance and location of the OOI. However, it was used in an unsupervised setting where one has no prior knowledge about the label (object vs. background) of each image patch. In our case, the area probabilities estimated by S-MILBoost provides information that we could use as prior knowledge.

The topic-motion model was designed for the case where at most one OOI appears in each frame. But this is not a problem for our system, because as long as one of the possibly many OOIs is discovered, the frame probability will be high. In other words, we don't need to identify every OOI to decide if a frame is relevant or irrelevant. Also notice that discovering the OOI is not our ultimate goal.

Denote frame k as d_k, where k indices over all frames. Each patch in d_k is associated with a visual word w, a position $\mathbf{r}$, and a hidden variable $z \in \{z_+, z_-\}$. Define $p(z_+|d_k)$ as the probability of a patch being originated from the OOI in frame k, and likewise $p(z_-|d_k)$ for the background. We define a spatial distribution $p(\mathbf{r}|z_+, d_k)$ that models the location of the patches originated from the OOI. We assume $p(\mathbf{r}|z_+, d_k)$ follows a Gaussian distribution, but other distributions (such as a mixture of Gaussians) could be used as well. Likewise, $p(\mathbf{r}|z_-, d_k)$ models the location of patches originated from background and we assume it follows a uniform distribution. The third distribution is $p(w|z_+)$, which models the appearance of the OOI. It is the normalized histogram over visual words corresponding to patches originated from the OOI. Likewise, $p(w|z_-)$ models the appearance of the background. We assume that the joint distribution of word w, position $\mathbf{r}$, and hidden label z of a patch in frame d_k is modeled as $p(z, \mathbf{r}, w|d_k) \equiv p(z|d_k)p(\mathbf{r}|z, d_k)p(w|z)$.

Define the state $\mathbf{s}(k)$ as the unknown position and velocity of the OOI in frame d_k. We assume a constant velocity motion model and the state evolves according to $\mathbf{s}(k+1) = \mathbf{F}\mathbf{s}(k) + \boldsymbol{\xi}(k)$, where $\mathbf{F}$ is the state matrix and the process noise

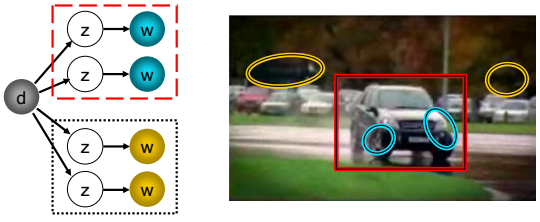


Fig. 5. Graphical model representation. Dashed lines are not the typical plate representation.

sequence $\boldsymbol{\xi}(k)$ is white Gaussian. Suppose at time k there are a number of m_k patches. If a patch is originated from the OOI, then its position can be expressed as $\mathbf{r}_i(k) = \mathbf{H}\mathbf{s}(k)+\boldsymbol{\zeta}_i(k)$, where $\mathbf{H}$ is the output matrix and the observation noise sequence $\boldsymbol{\zeta}_i(k)$ is white Gaussian; otherwise, the position is modeled as a uniform spatial distribution. The state estimate can be written as $\hat{\mathbf{s}}(k) = \sum_{i=1}^{m_k} \hat{\mathbf{s}}_i(k)\beta_i(k)$, where $\hat{\mathbf{s}}_i(k) = \hat{\mathbf{s}}(k^-) + \mathbf{W}(k)\boldsymbol{\epsilon}_i(k)$ is the updated state estimate conditioned on the event that $\mathbf{r}_i(k)$ is originated from the OOI, where $\boldsymbol{\epsilon}_i(k) = \mathbf{r}_i(k) - \hat{\mathbf{r}}(k^-)$ is the innovation, $\hat{\mathbf{r}}(k^-)$ is the observation prediction, $\hat{\mathbf{s}}(k^-)$ is the state prediction, and $\mathbf{W}(k)$ is the Kalman Filter gain [3]. The state estimation equations are essentially the same as in the PDA filter [3]. The association probability $\beta_i(k)$ is defined as $\beta_i(k) \propto N(\boldsymbol{\epsilon}_i(k)|0, \boldsymbol{\Upsilon}(k))p(z_i(k)|w_j, \mathbf{r}_i(k), d_k)$, where the first term contains motion information, the second term contains appearance and location information, and $\boldsymbol{\Upsilon}(k)$ is the innovation covariance.

Parameter Estimation. The distributions $P(w|z)$, $P(z|d)$, and $P(\mathbf{r}|z,d)$ are estimated using the Expectation-Maximization (EM) algorithm [6], which maximizes the log-likelihood $\mathcal{R} = \sum_k \sum_j \sum_i n_{kji} \log p(d_k, w_j, \mathbf{r}_i(k))$, where $n_{kji} \equiv n(d_k, w_j, \mathbf{r}_i(k))$ is a count of how many times a patch in d_k at position $\mathbf{r}_i(k)$ has appearance w_j. The EM algorithm consists of two steps. The E-step computes the posterior probabilities for the hidden variables:

$$p(z_l|d_k, w_j, \mathbf{r}_i(k)) = \frac{p(z_l|d_k)p(w_j|z_l)p(\mathbf{r}_i(k)|z_l, d_k)}{\sum_R p(z_l|d_k)p(w_j|z_l)p(\mathbf{r}_i(k)|z_l, d_k)} \tag{1}$$

The M-step maximizes the expected complete data likelihood. We adopt a Bayesian approach to estimating the probabilities, using m-probability-estimation [5]. First, notice that the *area probability*, $p_{k,a}$, computed from S-MILBoost contains prior knowledge about the OOI. This prior knowledge should be incorporated into the detection of temporally consistent candidate areas. This is a significant improvement over the algorithm in [11], which was completely unsupervised.

Noticing that each patch can belong to multiple candidate areas, we define the *patch probability* as the largest *area probability* among the candidate areas that cover an image patch. The *patch probability* is written as $p_{MIL}(z_l|d_k, w_j, \mathbf{r}_i(k))$, with the subscript "MIL" emphasizing that this probability is estimated from the outcome of

S-MILBoost. A simplified graphical model is illustrated in Fig. 5, where the variable $\mathbf{r}$ is omitted to simplify illustration. Dashed lines indicate groups of image patches having the same value of p_{MIL}. More specifically, dashed lines in red correspond to the red box (candidate area) in the picture, and blue (yellow) nodes in the graphical model correspond to blue (yellow) ellipses in the picture. We then obtain:

$$p(z_l|d_k) = \frac{\sum_{j,i} n_{kji} p_{MIL}(z_l|d_k, w_j, \mathbf{r}_i(k)) + \sum_{j,i} n_{kji} p(z_l|d_k, w_j, \mathbf{r}_i(k))}{\sum_{l,j,i} n_{kji} p_{MIL}(z_l|d_k, w_j, \mathbf{r}_i(k)) + \sum_{l,j,i} n_{kji} p(z_l|d_k, w_j, \mathbf{r}_i(k))} \quad (2)$$

$$p(w_j|z_l) = \frac{\sum_{k,i} n_{kji} p_{MIL}(z_l|d_k, w_j, \mathbf{r}_i(k)) + \sum_{k,i} n_{kji} p(z_l|d_k, w_j, \mathbf{r}_i(k))}{\sum_{j,k,i} n_{kji} p_{MIL}(z_l|d_k, w_j, \mathbf{r}_i(k)) + \sum_{j,k,i} n_{kji} p(z_l|d_k, w_j, \mathbf{r}_i(k))} \quad (3)$$

$$p(\mathbf{r}_i(k)|z_+, d_k) = \mathcal{N}(\mathbf{r}_i(k)|\hat{\mathbf{r}}(k), \boldsymbol{\Sigma}_{d_k}) \quad (4)$$

where $z_l \in \{z_+, z_-\}$ is the value taken by $z_i(k)$ and $\hat{\mathbf{r}}(k) = \mathbf{H}\hat{\mathbf{s}}(\mathbf{k})$ is the position estimate. The covariance $\boldsymbol{\Sigma}_{d_k}$ in the Normal distribution in Eq.(4) is the weighted covariance matrix of the observations $\mathbf{r}_i(k)$. The weighted covariance matrix is the covariance matrix with a weighted mass for each data point, with weights equal to the association probabilities $\beta_i(k)$. As a result, if the association probabilities have high uncertainty, the spatial distribution $p(\mathbf{r}|z_+, d)$ will be flatter; if low uncertainty, it will be sharper around the position of the OOI.

Finally, we propose a number of temporally consistent candidate areas that have $\hat{\mathbf{r}}(k)$ as center and with various sizes, as shown in Fig. 3(c). We use a 1.2 scale ratio between two areas, with the smallest one equal to the variance specified by $\boldsymbol{\Sigma}_{d_k}$ in Eq.(4), and with no more than 5 areas in total. Using various sizes is to increase system robustness in case of inaccurate size estimates.

5 Experiments

We use 15 video clips from YouTube.com and TRECVID [1]. Sample frames are shown in Fig. 6. Most of the clips are commercial advertisements with a well defined OOI and range from 20 to 356 seconds in length. We sample each video at two frames per second. In total, there are 3128 frames of size 320×240. The frames have visible compression artifacts.

The video frames are ground-truthed as positive or negative according to whether they contain the OOI; e.g., in a PEPSI commercial, we assume the PEPSI logo is the OOI. Each video clip is run twenty runs, where in each run we randomly select N_p frames from the positive frames and N_n frames from the negative frames as labeled data, where N_p and N_n are one or three. The rest of the frames are treated as unlabeled data. Results are averaged over the twenty runs. Notice that the labeled frames are labeled at the frame-level but not pixel-level.

Table 1 shows the average precision (area under precision-recall curve) of different methods. In the following, we will introduce the different comparative methods listed in Table 1 while we discuss the results. In general, we have the following observations:

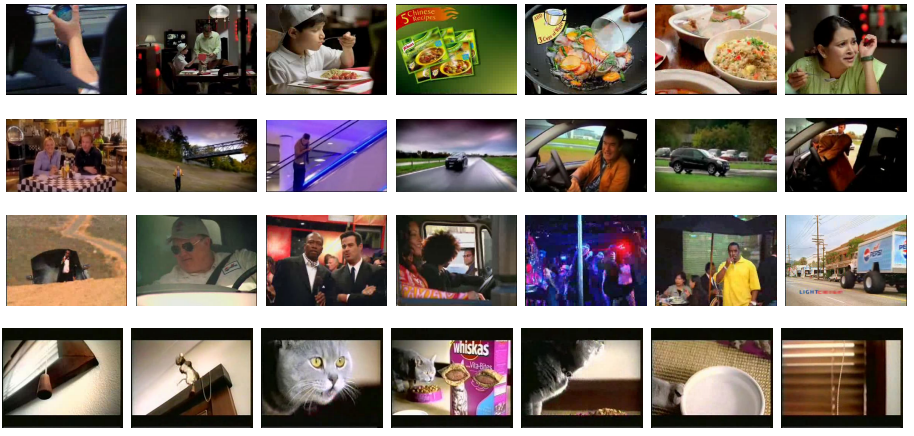


Fig. 6. Sample frames. Name of video clip, from top to bottom: Knorr, Benz, Pepsi, Whiskas.

Method 1: Supervised learning using only labeled data is consistently outperformed by the semi-supervised variants. When the number of labeled frames is low, its performance is close to by chance.

Method 2: Semi-supervised learning at frame level performs only marginally better than supervised learning when the number of labeled frames is as low as $(1+, 1-)$, but improves significantly as the number of labeled frames increases.

Method 3: Semi-supervised learning at sub-frame level with random areas consistently outperforms semi-supervised learning at the frame level. This justifies our claim in Sec. 4 that frame-level learning can be hindered when background clutter dominates the appearance features. Using sub-frames (candidate areas) helps the learning process to focus on the features originated from the OOI. The candidate areas consist of rectangles of size 160×120 with equal spacing between each other. In addition, a rectangle of size 320×240 covering

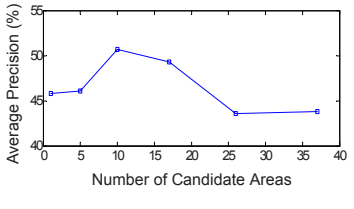


Fig. 7. Increasing the number of areas does not lead to increase in performance

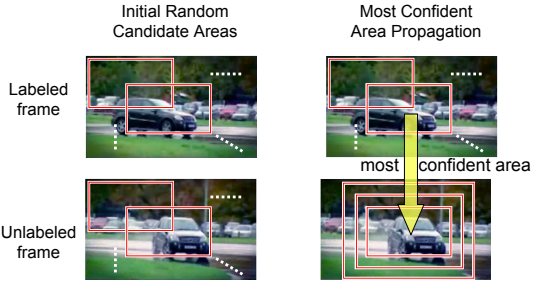


Fig. 8. Illustration of Method 4

the whole frame is used in here, in Method 4, and in the proposed method, in order to take care of large objects and inaccurate size estimates. After training S-MILBoost, we did not refine the placing of candidate areas, as we do in Method 4 and in the proposed method.

We experimented with different numbers of rectangles by changing the spacing between them and obtained different performances as shown in Fig. 7. There is a sweet spot at the number of 10 areas, which shows that the more candidate areas does not necessarily yield better performance. Even though increasing the number of areas will increase the chance that one of the candidate areas faithfully represents the OOI, the chance of overfitting also increases, hence the drop in performance. We also experimented with placing the areas more concentrated around the center of the frame but obtained similar results.

Table 1. Comparing the average precision (%). The number of labeled frames are one positive (1+) and one negative (1−) in the upper row, and three positives and three negatives in the lower row for each video sequence.

Sequence	Label	By Chance	Method 1 Supervised	Semi-Supervised			
				Method 2 Frame Level	Sub-Frame Level		
					Method 3	Method 4	Proposed
Benz	1+,1-	32.6	26.0	28.9	31.7	29.3	38.7
	3+,3-	32.5	29.1	52.9	54.6	48.8	58.3
Pepsi	1+,1-	34.1	32.6	34.3	41.9	39.1	42.3
	3+,3-	33.7	39.4	53.9	57.7	50.6	63.2
Whiskas	1+,1-	43.7	49.0	54.2	62.6	64.1	65.3
	3+,3-	43.5	53.9	71.2	77.0	78.1	73.8
SkittlesFunny	1+,1-	3.9	2.8	5.2	10.7	10.7	6.5
	3+,3-	2.0	4.1	11.3	21.2	22.5	22.7
CleanClear	1+,1-	21.2	14.4	15.5	45.4	41.8	36.1
	3+,3-	19.4	21.1	41.9	51.4	57.6	62.2
CatFood	1+,1-	39.0	40.5	41.7	62.7	65.1	66.9
	3+,3-	38.2	58.2	76.0	91.4	91.4	91.4
E-Aji	1+,1-	27.1	26.5	27.0	31.4	29.9	36.0
	3+,3-	25.5	23.8	32.6	42.7	34.7	36.2
CaramelNut	1+,1-	25.9	39.3	53.7	67.6	58.9	58.9
	3+,3-	24.1	58.4	67.5	67.6	70.2	70.2
Knorr	1+,1-	20.7	20.4	32.2	44.2	62.1	59.4
	3+,3-	18.5	20.2	48.9	57.2	69.4	67.7
Kellogs	1+,1-	18.4	19.8	20.0	26.4	30.3	30.3
	3+,3-	14.7	18.6	22.1	25.3	36.4	38.0
FlightSimul	1+,1-	10.8	15.4	43.5	42.6	53.5	59.6
	3+,3-	10.5	18.7	50.7	44.4	40.8	62.1
SpaceShuttle	1+,1-	4.8	2.8	2.8	3.5	3.7	4.2
	3+,3-	4.2	3.6	12.7	27.7	27.3	25.1
WeightAero	1+,1-	11.6	8.5	38.1	27.8	33.9	44.9
	3+,3-	11.2	46.9	56.3	40.9	48.5	56.1
WindTunnel	1+,1-	24.1	14.7	15.0	36.1	33.8	35.2
	3+,3-	23.8	41.6	47.2	56.9	56.7	56.1
Horizon	1+,1-	11.2	15.8	18.4	22.6	28.3	34.1
	3+,3-	10.5	18.0	41.3	44.1	48.9	54.6
Average	1+,1-	**21.9**	**21.9**	**28.7**	**37.1**	**39.0**	**41.2**
	3+,3-	**20.8**	**30.4**	**45.8**	**50.7**	**52.1**	**55.8**

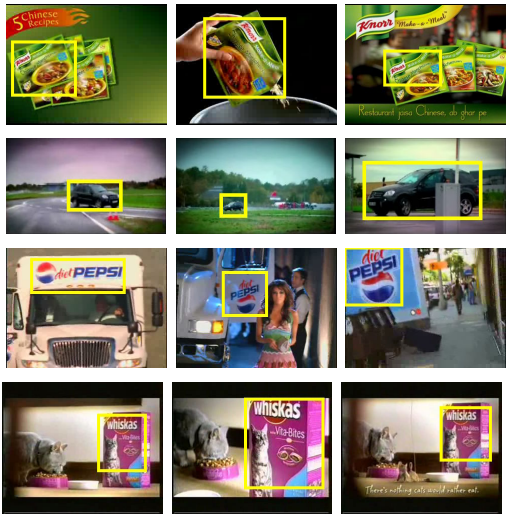


Fig. 9. Sample frames that are inferred as positive. A yellow box shows the candidate area with highest area probability. Name of video clip, from top to bottom: Knorr, Benz, Pepsi, Whiskas.

Method 4: Most confident area propagation: This method is the closest to the proposed method. Instead of using 'weak' tracking, we assume the OOI is stationary within a shot. As illustrated in Fig. 8, each unlabeled frame obtains its 'base' candidate area by replicating, from the nearest labeled frame, the size and position of the most confident area. Nearness can be defined as the visual similarity between frames or as the time difference between frames. We found the latter to work better. The base area is then resized and replicated within the frame using a 1.2 scale ratio between two areas, with the smallest one equal to the size of the base area, and no more than 5 areas in total. Since videos often contain multiple scene transitions or shots, we only allow the replication to happen within a shot and not across shots. If there are no labeled frames within a shot, we place random candidate areas in that shot.

In summary, the proposed method outperforms all the other methods (Table 1). Together with Fig. 7, this justifies our earlier expectation that properly placed candidate areas are crucial to the performance; using a huge number of candidate areas overfits the data and lowers the performance. The temporally consistent candidate areas reduce the need for a large number of uninformative candidate areas. Finally, in Fig. 9, we display some frames that are inferred by the proposed method.

6 Conclusion and Future Work

We have presented an approach for removing irrelevant frames in a video by discovering the object of interest. Through extensive experiments, we have shown that this is not easily achieved by directly applying supervised or semi-supervised learning methods in the literature developed for still images.

On a higher level, our method can be considered as a tracking system but without manual track initialization; the system finds out itself what the "best track" is, with the objective of agreeing with the user's labeling on which frames contain the object of interest.

References

1. http://www-nlpir.nist.gov/projects/trecvid/
2. http://www.robots.ox.ac.uk/~vgg/research/affine/
3. Bar-Shalom, Y., Fortmann, T.: Tracking and Data Association. Academic Press, London (1988)
4. Bennett, K., Demiriz, A., Maclin, R.: Exploiting unlabeled data in ensemble methods. Intl. Conf. Knowledge Discovery and Data Mining (2002)
5. Cestnik, B.: Estimating probabilities: A crucial task in machine learning. In: Proc. European Conf. Artificial Intelligence, pp. 147–149 (1990)
6. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. Royal Statistical Society 39, 1–38 (1977)
7. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE Trans. PAMI 28(4), 594–611 (2006)
8. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: British Machine Vision Conference (2002)
9. Julesz, B.: Textons, the elements of texture perception and their interactions. Nature 290, 91–97 (1981)
10. Li, Y., Li, H., Guan, C., Chin, Z.: A self-training semi-supervised support vector machine algorithm and its applications in brain computer interface. IEEE Intl. Conf. Acoustics, Speech, and Signal Processing (2007)
11. Liu, D., Chen, T.: A topic-motion model for unsupervised video object discovery. In: IEEE Conf. Computer Vision and Pattern Recognition (2007)
12. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Intl. J. Computer Vision 60, 91–110 (2004)
13. Mason, L., Baxter, J., Bartlett, P., Frean, M.: Boosting algorithms as gradient descent. In: Proc. Advances in Neural Information Processing Systems (NIPS) (1999)
14. Nigam, K., Ghani, R.: Analyzing the effectiveness and applicability of co-training. In: Intl. Conf. Information and Knowledge Management (2000)
15. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, Inc., San Francisco (1988)
16. Pritch, Y., Rav-Acha, A., Gutman, A., Peleg, S.: Webcam synopsis: Peeking around the world. In: IEEE Intl. Conf. Computer Vision (2007)
17. Rosenberg, C., Hebert, M., Schneiderman, H.: Semi-supervised self-training of object detection models. In: IEEE Workshop on Applications of Computer Vision (2005)
18. Schneiderman, H., Kanade, T.: Object detection using the statistics of parts. Intl. J. Computer Vision 56, 151–177 (2004)
19. Sivic, J., Schaffalitzky, F., Zisserman, A.: Object level grouping for video shots. Intl. Journal of Computer Vision 67, 189–210 (2006)
20. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: IEEE Intl. Conf. Computer Vision (2003)
21. van de Weijer, J., Schmid, C.: Coloring local feature extraction. In: Proc. European Conf. Computer Vision (2006)
22. Viola, P., Platt, J., Zhang, C.: Multiple instance boosting for object detection. In: Proc. Advances in Neural Information Processing Systems (NIPS) (2005)

ASN: Image Keypoint Detection from Adaptive Shape Neighborhood

Jean-Nicolas Ouellet and Patrick Hébert*

Computer Vision and Systems Laboratory
Laval University
Quebec, QC, Canada, G1V 0A6
{jouellet,hebert}@gel.ulaval.ca

Abstract. We describe an accurate keypoint detector that is stable under viewpoint change. In this paper, keypoints correspond to actual junctions in the image. The principle of ASN differs fundamentally from other keypoint detectors. At each position in the image and before any detection, it systematically estimates the position of a potential junction from the local gradient field. Keypoints then appear where multiple position estimates are attracted. This approach allows the detector to adapt in shape and size to the image content. One further obtains the area where the keypoint has attracted solutions. Comparative results with other detectors show the improved accuracy and stability with viewpoint change.

1 Introduction

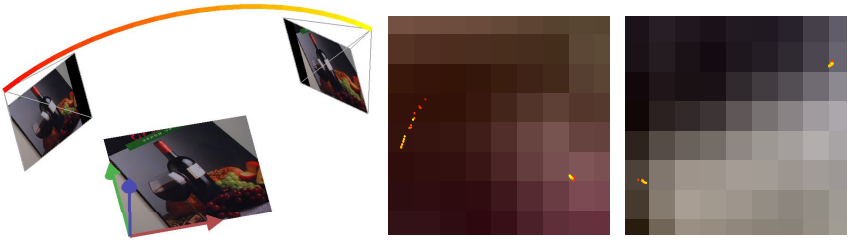
Recovering the accurate position of a camera from visual features usually involves placing artificial targets in the scene to ensure the detection and accurate localization of reliable keypoints from multiple viewpoints. To improve flexibility, natural features that are present in the scene should contribute. A critical aspect is the selection and position estimation of such reliable image keypoints especially when their number is limited. Each keypoint is important and no optimization procedure will be adequate unless sets of image keypoints accurately correspond to the same physical points in the scene.

Interest point, or keypoint, detection has been widely studied over the last 30 years and several detection approaches have been proposed. These points are distinctive locations in the image such as corners, other types of junctions and blob regions. A good review is presented in [1] where the authors study the repeatability of an observation under several types of transformations including severe scale and photometric changes.

In the last 10 years, significant progress has been accomplished in coupling multiscale feature detectors with descriptors. This has been motivated by finding point correspondences between two images or more generally in model-based recognition [2]. From the literature, it appears there is currently no universal detector/descriptor but interestingly, a combination of complementary operators seems to be a reasonable solution. While rich feature point descriptors will help in obtaining good initial correspondences, accurate keypoints that are only recognizable locally based on their seldomness will be used for improving the accuracy. We believe there is no compromise to be made

* The authors are grateful for support from NSERC Canada.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 454–467, 2008.



(a) (b) (c)

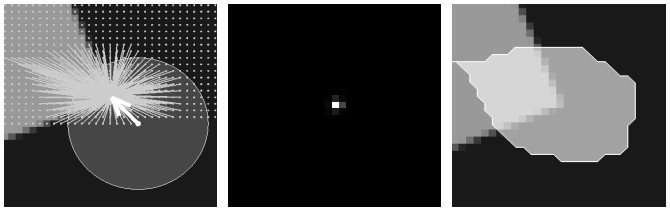
Fig. 1. Tracking the displacement of image features with viewpoint. a) Image features are detected along a trajectory over a rotation of 90 degrees. A dot is marked in the planar scene with its color corresponding to the camera position along the trajectory. b),c) Points backprojected onto the scene for the extracted SIFT and ASN approaches respectively.

between obtaining keypoints associated with discriminant local descriptors and accurate keypoints. When the number of accurate keypoints is not sufficient, one will have to add artificial targets. In this context, we focus on the localization accuracy of keypoints under viewpoint change. Although there may be severe viewpoint changes due to rotation, severe image scale changes are less an issue in this work. Moreover, in order to decouple the effect of occlusion, our analysis is limited to planar scenes.

An illustrative example is presented in Fig.1. While some SIFT features [3] are very stable, as shown on the right of Fig.1(b), others, such as the point on the left, will exhibit a gradual displacement reaching a few pixels as the camera moves along a trajectory. The displacements observed are related to the local image structure configuration which affects the stability under perspective distortion. In this simulation with a real image, the detected points are mapped in the fronto-planar view. The colors of the feature points correspond to the camera position along the trajectory. For applications like object recognition, the presence of such a displacement may not significantly affect the performance or quality of the results. However, this is not the case when the features are used to recover the exact geometry of the scene.

The classic direct approach for identifying image keypoints consists in applying a detector that evaluates a function within the neighborhood of each pixel in an image. Then, local extrema are identified, validated and their position can be refined based on the function value, for subpixel precision. The function is typically based on the local estimation of the second moment matrix of the gradient or the Hessian matrix [4,5]. The relative accuracy of this type of detectors was studied for an ideal corner model [6,7]. Nevertheless, in images of real scenes the result will be affected by the presence of several structures within a neighborhood. Moreover, when varying the viewpoint, the projective transformation will affect the distance between local structures in the image and an estimate based on a fixed size and shape neighborhood will be affected. Methods for adapting the neighborhood have been proposed for improving descriptors but this does not improve accuracy [4,8]. For localization accuracy, the shape of the neighborhood should reflect the image content.

We propose to reconsider the detection scheme. Instead of applying a detector, we systematically estimate the point of convergence of the local gradient field within the



(a) (b) (c)

Fig. 2. The ASN detector identifying poles and support map. a) The estimates are obtained at each image position and stored in an accumulator. The final state of the accumulator is depicted in (b) and the corresponding support map is displayed in (c).

neighborhood of each pixel in the image. These potential keypoints can be located several pixels away from the center of their estimation neighborhoods. As the detector processes the image, certain positions will attract more estimates than others, resulting in the formation of poles. The strength of a pole is the size of the area of attraction. The position of strong poles are then extracted as keypoints. Fig.2 illustrates this idea.

This approach is advantageous for three reasons. First, the area for estimating a keypoint adapts in shape based on the image content and needs not to be centered nearby the keypoint. Secondly, although it is possible to further characterize the strength of a pole, the detection is robust to the presence of nearby structures and is fundamentally independent of the image contrast. Finally, the area of attraction may provide further insight on the keypoint interest.

In section 2, definitions are provided and the approach is motivated from the most related works. In section 3, the method and the corresponding algorithm are developed. It is also shown how to apply multiple window scales and how to characterize and validate poles. Experiments follow in section 4 where it is shown that the extracted keypoints are highly stable even with strong viewpoint changes.

2 Keypoint Detection and Localization from the Gradient Field

Currently, most research efforts on keypoint detection focus on the correspondence between images, based on local information. A keypoint is located at the center of a region that is rich for description. It is then important to normalize scale and local deformation due to viewpoint. From the work of Lindeberg [9], who introduced the automatic scale selection to more recent works of Lowe on Scale Invariant Feature Transform (SIFT), Triggs [10] who generalizes the Harris operator to provide repeatable scale and orientation, Mikolajczyk and Schmid [4] who put forward the Harris (Hessian) affine detector, Tuytelaars and Van Gool [8] who proposed edge-based and intensity based detectors, the literature is abundant. Other detectors such as the MSER and salient region detector [11,12] are also worth noting. Good reviews on the detectors and descriptors have been published [1,13] along with other comparisons and evaluations [14,15].

In this paper we focus on local keypoint detectors. Among these detectors, most of the methods are based on the local measurement of the image Hessian matrix or the second moment matrix of the gradient. Analytical studies for the accuracy of ideal isolated

corners can be found in [6,7]. Among these detectors, the Forstner operator including variants such as the well known Harris detector are based on the second moment matrix [16,17]. They are particularly interesting since they make it possible to detect various types of keypoints such as L-corners, Y, T, and X junctions. A short analysis of these classic detectors will motivate the new approach proposed in this paper.

Given a point $\mathbf{x} = (x, y)$ in an a image I, the gradient $\nabla I(x, \sigma_d)$ at derivation scale σ_d is defined as $\nabla I(x, \sigma_d) = (I_x, I_y)^T$. The components I_x and I_y are obtained by convolving the Gaussian first order derivative $\frac{\partial}{\partial x} g(\sigma_d)$ with the image I at point $\mathbf{x}$ along the x and y directions respectively. The gradient field is the set $\mathcal{G} = \{\nabla I(\mathbf{x}, \sigma_d)\}$ for all $\mathbf{x}$ in the image. The second moment matrix $\Gamma(\mathbf{x}, \sigma_d)$ in $\mathbf{x}$ at derivation scale σ_d is defined as follows

$$\Gamma(\mathbf{x}, \sigma_d) = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}. \tag{1}$$

In order to obtain a local estimate $\overline{\Gamma}$ of Γ, the rotationally symmetric Gaussian function with scale σ_I is typically used for integrating the local gradient field components at point $\mathbf{x}_0 = (x_0, y_0)$

$$\overline{\Gamma}(\mathbf{x}_0, \sigma_d, \sigma_I) = \int \Gamma(\mathbf{x}, \sigma_d) g(\mathbf{x} - \mathbf{x}_0, \sigma_I) d\mathbf{x}. \tag{2}$$

In practice, the integral can be computed over a limited neighborhood window $N(\mathbf{x}_0, s)$ with center $\mathbf{x}_0$ and size s. From the 2×2 symmetric matrix $\overline{\Gamma}(\mathbf{x}_0, \sigma_d, \sigma_I)$, the eigenvalues can be analyzed or more efficiently, a combination of the trace and determinant of this matrix makes it possible to detect a keypoint by simply applying a threshold on the obtained scalar value. This operation is used for validating the presence of a keypoint in several detectors. For those detectors based on the Hessian matrix $\mathcal{H}$, the principle is the same except that $\mathcal{H}$ is computed instead of Γ. The maximum values can be further interpolated for subpixel precision of the position.

In [16], Forstner proposes a different way for refining the position estimate $\mathbf{x}_0$. To do so, it is proposed to minimize the weighted sum of the squared distances $d(\mathbf{x}_0, \mathbf{x})$ of the reference points $\mathbf{x}_0$ to the line passing through $\mathbf{x}$ in the direction orthogonal to $\nabla I(x, \sigma_d)$, where the weights are the squared gradient magnitude

$$\widehat{\mathbf{x}}_0 = \underset{\mathbf{x}_0}{\operatorname{argmin}} \int d^2(\mathbf{x}_0, \mathbf{x}) \left\| \nabla(\mathbf{x}, \sigma_d) \right\|^2 g_{\sigma_I}(\mathbf{x} - \mathbf{x}_0) d\mathbf{x}. \tag{3}$$

One can easily assess the benefit of this estimation step when observing an L-corner from different viewpoints. Actually, the opening angle will vary in the image. This behavior was studied by Rohr [6] for the raw detectors without refinement. Fig.3(a) shows the improvement after refinement. This refinement is particularly interesting since the keypoint location needs not to be at the center of the estimation window N. In practice, we have observed displacements higher than a pixel.

Due to band-limited images and noise, the gradient field is disrupted nearby a junction. Fig.4 illustrates the tangent field for both X and L types of junctions. Here, the tangent field is simply the set of vectors that are orthogonal to the gradient. The tangent field is parallel to edge directions. In Fig.4 the color encodes the gradient magnitude.

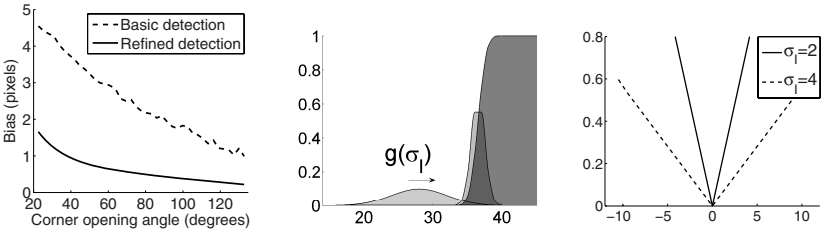


Fig. 3. a) Distance between the detected point and the actual junction with respect to the corner's opening angle before and after refinement of the detection using the Forstner operator. b) Location bias introduced by the Gaussian weight. When not exactly centered on the edge, the Gaussian imposes asymmetric weights on each side of the edge. c) The resulting bias in pixel with respect to the distance between the Gaussian center and the solution.

When the tangent field is symmetric around a junction, the error of the position estimate will be significantly reduced especially when N is centered at the junction. This characteristic is exploited in photogrammetric targets. However, in real scenes where it is more likely to observe L-corner types of junctions, the error will depend more on the integration scale. The larger the scale s, the closer the estimate will be toward the junction. It is then advantageous to select a larger scale. Nevertheless, the more likely N will overlap other structures.

Before completing this section, we consider the effect of the Gaussian weight g_{σ_I} for the integration window at the refinement step. For L-corners it will bias the position toward the interior of the corner. Actually, for this type of corner the maximum response of all operators lies inside the corner [6]. It is easy to understand this effect in one dimension. For instance in Fig.3(b) a step edge is depicted along with its derivative response superimposed and centered on the edge. Then suppose the position of the edge is estimated based on the center of gravity of the derivative response but using an integration window with Gaussian weight g_{σ_I} that is offset with respect to the derivative response. In the depicted case, the estimate will obviously be biased toward the left. As shown in Fig.3(c), the bias varies with g_{σ_I} and the offset.

This effect is also observed in images. In Fig.5, one can observe the detection corresponding to the maximum response of the Forstner operator (red point) and the refined position (white point). Increasing the integration scale will reduce the bias (see

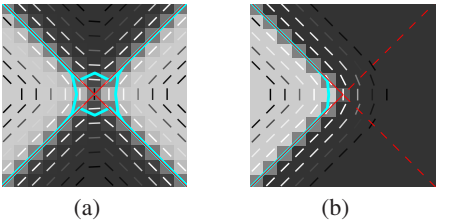
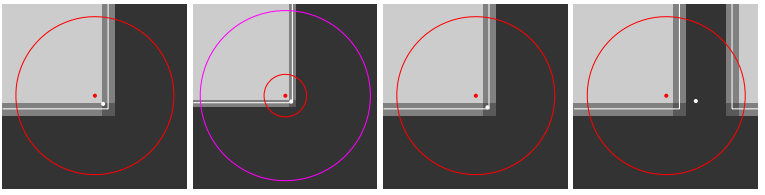


Fig. 4. Measured tangent field for a) an 'X' and b) an 'L' type of junction



(a) (b) (c) (d)

Fig. 5. Corner localization using a) the Forstner operator with $\sigma_d = 2, \sigma_I = 2$. In b) a larger integration scale is used $\sigma_d = 2, \sigma_I = 4$. c) The same window sizes as in (a) are used but without Gaussian weight for the integration window. d) Influence of a nearby structure.

Fig.5(b)). As expected, if the Gaussian weight is applied for the detection but not for the position refinement, the final result is unbiased (see Fig.5(c)). Nevertheless, that creates another problem depicted in Fig.5(d). The presence of nearby disrupting structures introduces a bias in the estimate. The Gaussian weight used in most keypoint operators greatly contributes to minimize this effect, especially for square windows. It will, however, introduce the bias especially for the common L-corners.

From these observations and analysis, we will retain three conclusions:

1. From the gradient (tangent) field, it is possible to estimate a local optimal position of any type of junctions such as X, T, Y and L-corners.
2. When the estimation is performed away from the keypoint position, a uniform weight in the integration window will prevent bias.
3. A larger integration scale leads to better estimates but is more prone to disrupting nearby structures.

3 Keypoint Detection Based on Pole Identification

Based on these considerations, we propose the following approach to detect and estimate the positions of the keypoints. In order to detect reliable keypoints, we search for poles in the image. There is a pole at a given position $\mathbf{x}_0$ in the image when multiple estimates from different positions of N are attracted at this position. The number of estimates at a pole is a local peak. The strength of a pole located at $\mathbf{x}_0$ is the number of window neighborhoods, $N(\mathbf{x}, s)$, that led to $\mathbf{x}_0$. It is therefore proposed to systematically apply the least-squares estimate of equation 3 at all positions $\mathbf{x}$ in the image and identify poles without any pre-detection. Significant poles are then validated and identified as significant keypoints. This approach is remarkably simple in its principle. However, it fundamentally differs from the process of searching for the position where a function of the second moment matrix or Hessian matrix leads to a maximum. Actually, a pole arises from the estimation of neighborhood windows at multiple positions. That is why it can adapt in shape to the presence of nearby structures.

In order to formalize this idea, we revisit the definition of an estimate from eq. 3 with

$$\widehat{\mathbf{p}} = F_N(\mathbf{x}, s) = \underset{\mathbf{p}}{\operatorname{argmin}} \sum_{y \in N(\mathbf{x},s)} d^2(\mathbf{p}, \mathbf{y}) \|\nabla(\mathbf{y}, \sigma_d)\|^2. \tag{4}$$

In this equation, we have replaced the integral by a discrete sum over the neighborhood $N(\mathbf{x}, s)$ and we have removed the Gaussian weight in eq. 3. This will eliminate the bias source described in section 2.

Let us also define the support map for a given position as

$$S(\mathbf{y}) = \{\mathbf{x}\} \mid \|F_N(\mathbf{x}, s) - \mathbf{y}\| < \epsilon. \tag{5}$$

$S(\mathbf{y})$ is the set of positions $\mathbf{x}$ whose estimates $\widehat{\mathbf{p}}$ fall in the vicinity of $\mathbf{y}$. There is a pole in $\mathbf{y}$ if the following condition is met

$$\#S(\mathbf{y}) > \#S(\mathbf{y'}),\ \forall\, \mathbf{y'} \ \mid\ \|\mathbf{y} - \mathbf{y'}\| < \epsilon, \tag{6}$$

where $\#$ stands for the cardinality. The exact subpixel position of the pole is estimated by averaging all estimates $\widehat{\mathbf{p}}$ that have been attracted by the pole. In the same way it is accomplished for validating keypoints based on the second moment matrix, it is possible to characterize the second moment matrix of the augmented support map S^+ of a pole. S^+ is defined as the union of all neighboring windows whose estimates have been attracted by the pole

$$S^+(\mathbf{y}) = \bigcup N(\mathbf{x}, s) \mid \mathbf{x} \in S(\mathbf{y}). \tag{7}$$

Fig.2 illustrates the progression of the estimation process along with the attracted estimates and the final support map of a detected pole. It is worth noting that the identification of a pole is not directly related to the junction contrast but to the area of its support map. Moreover, this process will improve robustness similar to Hough-based approaches where a consensus is sought locally [18]. However, we avoid the detection of numerous false points that must be discarded a posteriori [18], by searching for a concentration of independent estimates.

Despite its capability of adapting to the local structure, it is possible that a large integration scale makes it impossible to identify a pole since N always overlaps multiple structures in the image. It is thus mandatory to reduce the integration scale. In order to cope with this situation, the integration scale is reduced progressively. Poles are detected and validated at the largest scale for higher immunity to noise. Then, the scale is recursively decreased and valid poles are sought for at each scale. A small image area around a pole is deactivated such that no new pole can be accepted within the area at smaller scale. The whole procedure is simple and its detailed implementation is described in section 3.2.

3.1 Pole Validation

The number of estimates corresponds to the surface (cardinality) of a pole's support map, S, whose size and shape will vary with the type of junction and the presence of structures in the vicinity of the keypoint. Moreover, the probability that a junction be within the neighboring window $N(\mathbf{x}, s)$ is proportional to the area of the window. Thus a reasonable threshold for validating the strength of a pole is $\tau = 0.2\ Area(N(\mathbf{x}, s))$. This efficiently identifies significant poles while eliminating poles resulting from noise.

For stability, it is not sufficient to reject poles below τ. Unstable poles may appear along low curvature edges. The second moment matrix is computed over the extended support map S^+ and the ratio of its eigenvalues $\lambda_{max}/\lambda_{min}$ is tested. In all cases, the pole is rejected when the ratio of the eigenvalues is higher than 10. Such test is common in the literature; it eliminates an operator response over curved edges [3,17].

From the second moment matrix, one can further assess the uncertainty of the pole's position. For that purpose, let us reformulate the least-squares solution of eq. 4 into matrix form. The estimates result from the least-squares solution of $\mathbf{Kp} = b$

$$\mathbf{K} \equiv \begin{bmatrix} \frac{\partial I}{\partial x}_1 & \frac{\partial I}{\partial y}_1 \\ \vdots & \vdots \\ \frac{\partial I}{\partial x}_i & \frac{\partial I}{\partial y}_i \\ \vdots & \vdots \\ \frac{\partial I}{\partial x}_n & \frac{\partial I}{\partial y}_n \end{bmatrix}; b \equiv \begin{bmatrix} x_1 \frac{\partial I}{\partial x}_1 + y_1 \frac{\partial I}{\partial y}_1 \\ \vdots \\ x_i \frac{\partial I}{\partial x}_i + y_i \frac{\partial I}{\partial y}_i \\ \vdots \\ x_n \frac{\partial I}{\partial x}_n + y_n \frac{\partial I}{\partial y}_n \end{bmatrix}, \tag{8}$$

where n is the number of pixels in $N(\mathbf{x}, s)$. The solution of this system is given by

$$\widehat{\mathbf{p}} = F_N(\mathbf{x}, s) = \overline{\Gamma}^{-1}\mathbf{K}^T b, \tag{9}$$

with the second moment matrix $\overline{\Gamma} = \mathbf{K}^T\mathbf{K}$. A solution exists if $\overline{\Gamma}$ is invertible. The uncertainty of the solution can be obtained from the covariance matrix of the point estimate as the product of the error variance with $\overline{\Gamma}^{-1}$, $\mathbf{C}_p = \sigma_{err}^2 \overline{\Gamma}^{-1}$. The error variance is given by $\sigma_{err}^2 = \frac{\|\mathbf{Kp}-b\|^2}{n-2}$ [16]. From this uncertainty expression, one can further validate the precision of the poles with the variance of the estimation error, σ_{err}^2. The variance depends on the noise level as well as on the discrepancy of the local tangent field. As the support map overlaps other contrasted structures nearby, the variance increases whereas the uncertainty will remain low. Poles arising from the interaction with such structures that do not intersect at a single point, are referred to as virtual junctions. Virtual junctions are not located on actual junctions of the image and exist only for specific integration scale. To improve the quality of the points, we decouple σ_{err}^2 from $\mathbf{C_p}$ and eliminate poles for which σ_{err} is higher than a tolerance T_{err}.

3.2 Implementation

In order to obtain the tangent field values at each pixel, the image gradient is first evaluated by convolving the image with a Gaussian derivative filter with $\sigma_d = 1$. Next, using Eq. 9, the least-squares estimate is computed within the integration window $N(\mathbf{x}, s)$ for all positions in the image. If present, color channels are directly handled by the operator by combining the gradient vectors from the three channels. In our implementation, the image is processed three times with a circular integration window of decreasing radii $s = \{9, 6, 3\}$.

As the operator is applied at each image position, the estimates are stored in the cells of an accumulator. The accumulator's discretization is the same as the image; this provides a direct correspondence between cells and pixels. Since an estimate will generally

not map to the center of a cell, we increment the four closest cells by distributing the coefficients of a bilinear interpolation. This amounts to adding the value of a 2D normalized Gaussian function of $\sigma_A = 0.5$ centered on the estimate. The position $\mathbf{x}$ of N leading to a given estimate is also stored to allow fast extraction of the support map for a given pole. More precisely, we store the list of positions $\mathbf{x}$ at the rounded coordinates of their estimates $\widehat{\mathbf{p}}$ in a first table as $\mathbf{H}(\lfloor\widehat{\mathbf{p}}\rfloor) \leftarrow \mathbf{x}$ and, in a second table $\mathbf{E}$, we store each estimate $\widehat{\mathbf{p}}$ at the position at which it was estimated, $\mathbf{E}(\mathbf{x}) \leftarrow \widehat{\mathbf{p}}$.

After processing the image at a given integration scale s, the local maxima of the accumulator are identified. Positions whose support map contains a sufficient number of estimates ($> \tau$) are identified as potential poles. To extract the support map, one must identify the estimates mapping within $\epsilon = 2 * \sigma_A$ pixels of the local maxima.

The poles are finally re-estimated as the weighted mean of the estimates associated with its support map. A normalized Gaussian weighting function with ($\sigma = \sigma_A$) centered on the local maximum is used for this purpose. The poles are then validated according to the ratio $\lambda_{max}/\lambda_{min} < 10$ and $\sigma_{err} < 0.25$ described in the previous section. As the operator integration scales are explored, poles detected at larger integration scales are prioritized; we reject any local maximum obtained at smaller scales within a radius of $\rho = 2.5$ pixels of a valid pole. The complete process is summarized in algorithm 1.

Input: Image gradient
Output: Poles and Support Maps
foreach *Scale s in* $\{9, 6, 3\}$ **do**
  Initialize accumulator $\mathbf{A} \leftarrow \mathbf{0}$
  // Estimation
  foreach *image position* $\boldsymbol{x}$ **do**
    Estimate $\mathbf{p} \leftarrow f_N(\mathbf{x}, s)$
    if $\boldsymbol{p}$ *exists* **then**
      Increment the accumulator $\mathbf{A}(\mathbf{p})$
      Store $\mathbf{H}(\lfloor\mathbf{p}\rfloor) \leftarrow \mathbf{x}$ and $\mathbf{E}(\mathbf{x}) \leftarrow \mathbf{p}$
    end
  end
  $\Omega \leftarrow$ positions of local maxima in the accumulator $\mathbf{A}$
  // Validation
  Remove positions from Ω that are closer than ρ pixels from higher scale poles
  foreach *position* $\boldsymbol{x}$ *in* Ω **do**
    Extract the support map $S(\mathbf{x})$ using $\mathbf{H}$ and $\mathbf{E}$
    if $\#S(\boldsymbol{x}) > \tau$ **then**
      Re-estimate the pole position $\mathbf{p}$ from the estimates in $\mathbf{E}$, indicated by $S(\mathbf{x})$
      Compute $\overline{I}$ over $S^+(\mathbf{x})$
      if $(\lambda_{max}/\lambda_{min} < 10)\ and\ (\sigma_{err} < T_{err})$ **then**
        $Poles \leftarrow \{\mathbf{p}, S(\mathbf{x})\}$
      end
    end
  end
end

Algorithm 1. The ASN detector: poles and support maps extraction

4 Results

First, we present results with both synthetic and real images to illustrate how the operator can adapt in scale and shape to image content. Then, following the demonstration in Fig.1, we present comparative experiments on the displacement and repeatability of keypoints using virtual cameras to simulate different controlled viewpoints from real images. We compare three detectors: ASN, Forstner with refinement, and SIFT. The images are generated using homographies so the ground truth allows any bias to be identified while preventing matching errors.

4.1 Adaptation to Image Content

In order to illustrate the behavior of the detector, an image was synthesized with squares of different sizes and gaps in between. The corners of the squares, visible at the intersections of the white lines in Fig.6(a), are only detectable at specific scales. The support maps S detected by the ASN detector are superimposed on the image in Fig.6(a) along with the poles identified by white dots. The color of the regions encodes the integration scale at which a pole was identified (blue $s = 9$, green $s = 6$, red $s = 3$ pixels). The red dots mark the poles (virtual junctions) that were rejected due to high σ_{err} ($\sigma_{err} > 0,25$). The cases presented in Fig. 6(b) to 6(g) originate from real images. For these latter cases, S^+ is superimposed using color transparency.

A pole will be close to the center of S when there is no other structure in the vicinity. It is also worth noting that a pole is not confined to its support map (see Fig.6(a) and Fig.6(c)). This justifies the importance of estimating a position that may not correspond to the center of the integration window. However, if the image local structure interferes with the operator for all positions near a junction, the estimates might form a strong pole where no real junction exists. Such virtual poles can be observed as red dots in

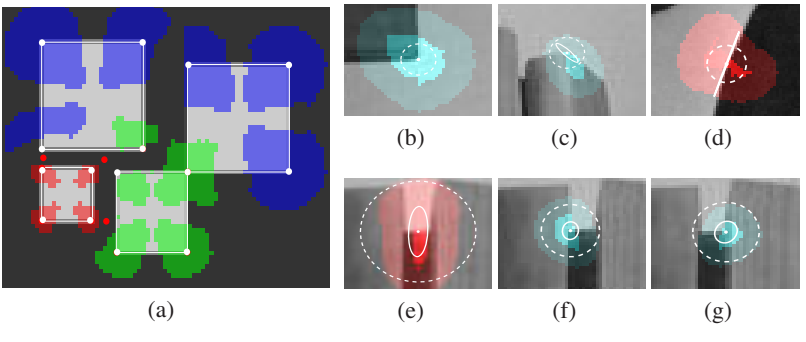


Fig. 6. a) Keypoints and support maps obtained at three different scales. The colors (red, green and blue) indicate the integration scale of the operator which is of 9, 6 and 3 pixels respectively. (b)-(g) Insets of keypoints obtained in real images. The associated uncertainty and RMS errors are depicted as white ellipses and dotted white circles (factor 100x) respectively. The keypoints' support maps, S, (strong color) along with their extensions, S^+ (light color) are superimposed. The red dots indicate a virtual junction.

the synthetic example of Fig.6(a), and also for a real image in Fig.6(e). These poles are sensitive to the scale of the operator. They are identified by the analysis of the RMS error obtained after the estimation. At smaller scales the real junction will eventually be identified. In the synthetic example of Fig.6(a), this resulted in the green and red regions corresponding to regions extracted at smaller scales. For real images, the virtual pole in Fig.6(e) was rejected and the real junctions in Fig.6(f) and Fig.6(g) were identified afterwards. Poles can also appear on low curvature edges as displayed in Fig.6(d). These less stable points do not pass the verification on the eigenvalue ratio.

4.2 Viewpoint Displacement and Repeatability

In this section we examine how viewpoint changes affect the keypoint detection and localization. For this purpose, a series of 81 images are produced after applying homographies from real images to exactly reproduce how this planar scene would be acquired from the positions depicted in Fig.7(a). In the figure, the virtual camera is oriented toward the scene center and moved radially from this point. More precisely, the camera covers a zenith[1] angle varying from -45 to +45 degrees with 5 degree steps and an azimuthal[2] angle varying from 0 to 135 degrees with 45 degree steps. The detector was tested on multiple scenes among which 3 different cases are presented here.

We compare the ASN detector with the SIFT and Forstner operators. When it is required, the image gradient is computed by the convolution of a Gaussian derivative of $\sigma_d = 1$. The integration scale of the Forstner operator is set to $\sigma_I = 3$. Both, the ASN and Forstner operators reject a keypoint when the eigenvalue ratio is 0.1. We used the SIFT implementation that is available online. For a given image, the ASN and Forstner operators find an equivalent number of keypoints while SIFT typically finds two to three times more keypoints.

Since we know the exact mapping between the images, we can align the points extracted between two images and measure any bias present. In this experiment, the points detected in each image are mapped to the front view image. This reference image is taken at a zenith angle of zero degree. A point is successfully matched if the actual homography between the images maps the detected point within a distance of 0.7 pixel of a point previously transferred to the reference image. This method is similar to tracking and allows a point to be matched over a large baseline while avoiding false matches.

We first calculate the mean repeatability of an operator at each zenith angle. The repeatability for a given image is obtained as the ratio of the number of matched points between the image and the reference image, to the number of detected points in the reference image. The mean repeatability for the three operators is plotted with respect to viewpoint in Fig.7. This figure shows equivalent repeatability between ASN and Forstner. The SIFT operator lower score is due to the large number of keypoints detected in each image.

Besides repeatability, we compute the maximum displacement of the points with respect to viewpoint, which indicates the capability of an operator to identify the same physical point. The displacement of a point is the norm of the vector between the point

[1] Measured from the Z-axis.
[2] Measured from the X-axis in the XY-plane.

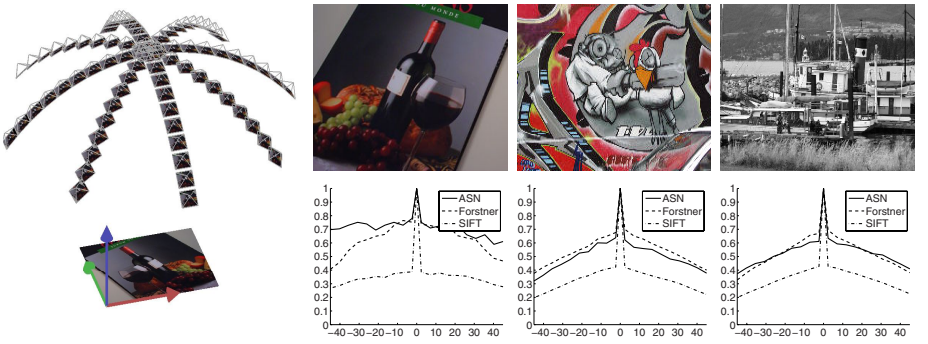


Fig. 7. a) Positions of the virtual camera where the images of the planar scene are acquired. (b)-(c)-(d) Repeatability under viewpoint variation for three different scenes. The results obtained for a given scene are displayed below the reference image. The horizontal axis is the zenith angle of the viewpoint, in degrees.

mapped from a given image to the reference image, and its correspondent detected in the reference image. Since no noise is added to the synthetic images, the displacement is caused by the estimation bias such as explained in previous sections. The maximum displacement of a point is computed over all azimuthal angles for a given zenith angle. The histograms in Fig.8 compile the point maximum displacements observed with respect to the zenith angle, from left to right for the ASN, Forstner and SIFT operators respectively. The white and black dotted lines depict the maximum and mean displacement values at a given zenith angle.

The highest displacement is observed for SIFT features. To explain such results we refer back to Fig.1(b). The SIFT operator detects region centers based on the Difference of Gaussian (DoG) function. However, SIFT keypoints are also detected near junctions. Actual keypoints arising from these situations are displayed in Fig.1(b). On the right, a stable keypoint is detected over a small region. The DoG function benefits from the rotational symmetry of the underlying structure. On the left, the keypoint is located near a junction in the image and is systematically shifted as the camera moves along the trajectory. The operator is more sensitive to non-symmetric regions. The latter situation is commonly encountered and explains the high displacement values for SIFT keypoints.

From the histograms in Fig.8, the maximum displacements of the keypoints extracted with the ASN operator are smaller with the viewpoint, regardless of the scene. The maximum displacements originate from image structures of low curvature and the detection of such situations from a single view is still a challenge. Interestingly, both ASN and Forstner operators detect the same number of points and exhibit similar repeatability (see the bottom row of Fig.7). Yet, as it can be observed in Fig.8, systematically for all scenes, the keypoints found by the ASN operator display smaller position variations. The difference arises from the adaptive estimates as well as the absence of Gaussian weight in the estimation window for the ASN detector.

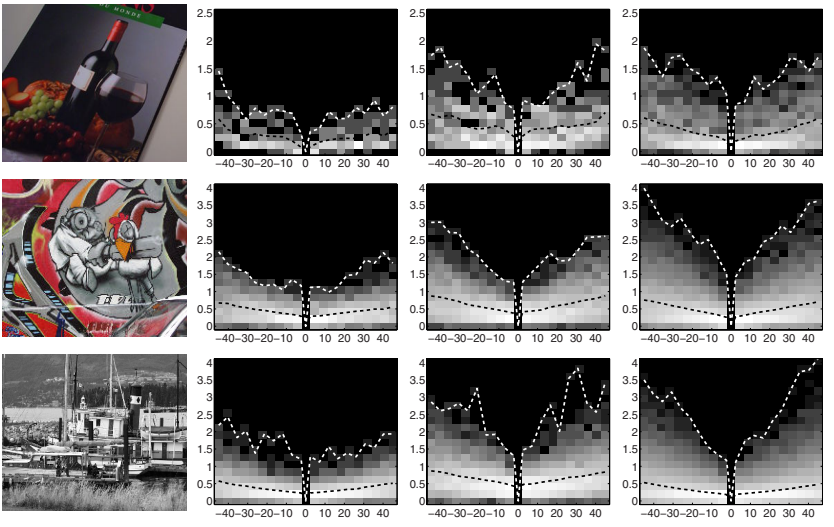


Fig. 8. Histograms of the maximum point displacements under viewpoint variation for three different scenes. The columns present vertical histograms of the maximum displacement (in pixels) with respect to the zenith angle (in degrees) for the ASN (b), Forstner (c) and SIFT (d) operators. In the histograms, the brightness is proportional to the logarithm of the number of points with displacement indicated on the vertical axis. The white and black dotted lines represent the maximum and mean displacement values at a given zenith angle.

5 Conclusion

The principle of the ASN detector differs fundamentally from other keypoint detectors. At each position in the image, it systematically estimates the position of a potential junction from the local gradient field in the image. Keypoints then appear where multiple position estimates are attracted. These are poles. Since a keypoint results from estimations at several positions of a window neighborhood, the detector adapts to image content and one obtains the area where the keypoint is a local landmark, namely the support map. It was further shown how the ASN detector adapts integration scale for identifying physical points. By prioritizing larger scales and avoiding Gaussian weight in the estimation, improved accuracy and stability with viewpoint change were shown. This type of keypoints will be useful to improve accuracy when combined with less accurate keypoints featuring rich descriptors. In future work, it will also be interesting to investigate the role of the support map for disambiguating correspondences.

References

1. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. International Journal of Computer Vision 37(2), 151–172 (2000)
2. Lowe, D.G.: Object recognition from local scale-invariant features. In: International Conference on Computer Vision, pp. 1150–1157 (1999)

3. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
4. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. International Journal of Computer Vision 60(1), 63–86 (2004)
5. Bay, H., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. In: European Conference on Computer Vision, pp. 404–417 (2006)
6. Rohr, K.: Localization properties of direct corner detectors. Journal of Mathematical Imaging and Vision 4(2), 139–150 (1994)
7. Deriche, R., Giraudon, G.: Accurate corner detection: an analytical study. In: International Conference on Computer Vision, pp. 66–70 (1990)
8. Tuytelaars, T., Gool, L.V.: Wide baseline stereo matching based on local, affinely invariant regions. In: British Machine Vision Conference, pp. 412–422 (2000)
9. Lindeberg, T.: Feature detection with automatic scale selection. International Journal of Computer Vision 30(2), 79–116 (1998)
10. Triggs, B.: Detecting keypoints with stable position, orientation, and scale under illumination changes. In: European Conference on Computer Vision, pp. 100–113 (2004)
11. Matas, J., Chum, O., Martin, U., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: British Machine Vision Conference, vol. 1, pp. 384–393 (2002)
12. Kadir, T., Zisserman, A., Brady, M.: An affine invariant salient region detector. In: European Conference on Computer Vision, pp. 228–241 (2004)
13. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. Pattern Analysis and Machine Intelligence 27(10), 1615–1630 (2005)
14. Fraundorfer, F., Bischof, H.: A novel performance evaluation method of local detectors on non-planar scenes. In: Computer Vision and Pattern Recognition, vol. 3, pp. 33–33 (2005)
15. Moreels, P., Perona, P.: Evaluation of features detectors and descriptors based on 3d objects. International Journal of Computer Vision 73(3), 263–284 (2007)
16. Forstner, W.: A framework for low level feature extraction. In: European Conference on Computer Vision, pp. 383–394 (1994)
17. Harris, C., Stephens, M.: A combined corner and edge detector. In: Alvey Vision Conference, pp. 147–151 (1988)
18. Park, S.J., Ahmad, M.B., Rhee, S.H., Han, S.J., Park, J.A.: Image corner detection using radon transform. In: International Conference on Computational Science and Applications, pp. 948–955 (2004)

Online Sparse Matrix Gaussian Process Regression and Vision Applications

Ananth Ranganathan[1] and Ming-Hsuan Yang[2]

[1] Honda Research Institute, Mountain View, CA 94041
aranganathan@honda-ri.com
[2] University of California, Merced, CA 95344
mhyang@ucmerced.edu

Abstract. We present a new Gaussian Process inference algorithm, called Online Sparse Matrix Gaussian Processes (OSMGP), and demonstrate its merits with a few vision applications. The OSMGP is based on the observation that for kernels with local support, the Gram matrix is typically sparse. Maintaining and updating the sparse Cholesky factor of the Gram matrix can be done efficiently using Givens rotations. This leads to an exact, online algorithm whose update time scales linearly with the size of the Gram matrix. Further, if approximate updates are permissible, the Cholesky factor can be maintained at a constant size using hyperbolic rotations to remove certain rows and columns corresponding to discarded training examples. We demonstrate that, using these matrix downdates, online hyperparameter estimation can be included without affecting the linear runtime complexity of the algorithm. The OSMGP algorithm is applied to head-pose estimation and visual tracking problems. Experimental results demonstrate that the proposed method is accurate, efficient and generalizes well using online learning.

1 Introduction

Learning regression functions from data has been an important problem in machine learning and computer vision with numerous applications. In recent years, kernel machines such as Support Vector Machines and Gaussian Processes have demonstrated great success in learning nonlinear mappings between high dimensional data and their low dimensional representations. For numerous vision applications where we have continuous stream of data, it is of great interest to learn nonlinear regression functions in an online manner.

In this paper, we propose a new Gaussian Process (GP) regression algorithm, called Online Sparse Matrix Gaussian Process (OSMGP) regression, that is exact and allows fast online updates in linear time for kernel functions with local support. This combination of exact inference and fast online updates is a novel contribution. We show that when the Gram matrix is sparse, as is the case when kernels with local support are used, an efficient representation is to maintain and update the Cholesky factor of the Gram matrix instead of the matrix itself. During online learning, when a new point is added to the training sequence, this introduces a new row and column into the Gram matrix. Instead of recomputing the Cholesky factor for the matrix, which would be expensive,

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 468–482, 2008.

we use Givens rotations to incrementally update it. Givens rotations are guaranteed to update the factorization in $O(n)$ time for a sparse matrix, where the Gram matrix has size $n \times n$, but can be much faster in practice.

We demonstrate that even though the Cholesky factor of the Gram matrix may become dense due to repeated applications of Givens rotations as training points are added, this can be overcome through the use of variable reordering, which restores sparsity of the matrix. If the hyperparameters of the GP are learned offline and not changed during online execution, the overall algorithm is linear. However, if these are learned automatically using a maximum likelihood method, this introduces a periodic quadratic update when the optimization is performed and the complete Gram matrix is recomputed.

As an additional contribution we propose the use of matrix downdating using hyperbolic rotations to also learn the hyperparameters of the GP in constant time. Hyperbolic rotations are used to incrementally recompute a matrix factorization when a row and column from the matrix are removed. This operation can be performed in $O(n)$ time similar to Givens rotations. Downdating introduces an approximation into the GP update but enables the Gram matrix to be maintained at a constant size by removing a training point from the training set whenever a new point is added. Thus, the size of the training set does not change. Hence, recomputing the Gram matrix after re-learning the hyperparameters can be done in constant time.

The proposed OSMGP algorithm is applied to head pose estimation and visual tracking problems. In particular, for head pose estimation, we use a dataset that is both comprehensive and exhaustive, containing 17 people and over 30,000 test images with ground-truth. Extensive comparisons with the existing methods show the accuracy, robustness and generality of our pose estimation system. For the tracking problem, we use publicly available datasets and demonstrate that the accuracy of the regression tracker is comparable to existing systems.

2 Gaussian Process Regression

A Gaussian Process is a distribution over the space of functions, which is usually defined as a collection of random variables, any subset of which have a Gaussian distribution. GPs can be viewed as probabilistic kernel machines, and hence, can provide not only a mean value prediction but also the uncertainty measured in terms of standard deviation for a test sample. A large standard deviation signals the absence of any training data in the neighborhood of the test sample, and provides an indication of poor generalization. A Gaussian Process is completely defined by a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$. A random function $f(\mathbf{x})$ distributed according to a GP is written as $f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right)$. The GP is transformed into a probabilistic kernel machine if we take the covariance function to be a semi-positive definite Mercer kernel, such that the covariance between points $\mathbf{x}_i$ and $\mathbf{x}_j$ is given by $k(\mathbf{x}_i, \mathbf{x}_j)$.

For performing regression, we assume the availability of n training inputs $X = \{\mathbf{x}_{1:n}\}$ and corresponding outputs $\mathbf{y} = \{y_{1:n}\}$. The covariance function of the GP is then given by the $n \times n$ Gram matrix $K(X, X) = K$. Typically, the kernel function has a number of parameters θ, which are also called the hyperparameters of the GP, that have

to be learned using the training set. For example, for the Radial Basis Function (RBF) kernel given as $k(\mathbf{x}_i, \mathbf{x}_j) = c \exp\left\{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{2\eta^2}\right\}$, the hyperparameters are $\theta = (c, \eta)$. Hyperparameter learning is usually done by maximizing the marginal log-likelihood

$$p(\mathbf{y}|X, \theta) = -\frac{1}{2}\log\left|K + \sigma^2 I\right| - \frac{1}{2}\mathbf{y}^T\left(K + \sigma^2 I\right)^{-1}\mathbf{y} - \frac{n}{2}\log 2\pi \tag{1}$$

where I is the identity matrix of same dimensions as K and σ is the standard deviation of additive Gaussian noise. This learning step is performed offline since it involves inverting a potentially large Gram matrix.

The regression-based prediction for a given test point $\mathbf{x}^\star$ is given by the conditional distribution on the test output given the training data and the test input. This is again a Gaussian distribution $p(\mathbf{y}^\star|X, Y, \mathbf{x}^\star) = \mathcal{N}(\mu^\star, \Sigma^\star)$ with the predictive mean and covariance given by

$$\mu^\star = \mathbf{k}^{\star\mathbf{T}}\left(K + \sigma^2 I\right)^{-1}\mathbf{y}, \quad \Sigma^\star = k(\mathbf{x}^\star, \mathbf{x}^\star) - \mathbf{k}^{\star\mathbf{T}}\left(K + \sigma^2 I\right)^{-1}\mathbf{k}^\star \tag{2}$$

where $\mathbf{k}^\star = [k(\mathbf{x}^\star, \mathbf{x}_1), k(\mathbf{x}^\star, \mathbf{x}_2), \ldots, k(\mathbf{x}^\star, \mathbf{x}_n)]$.

The GP regression algorithm, as described above, is not useful for applications with large datasets since it does not scale with the number of data points. Training is $O(n^3)$ in complexity by (1) as it involves inversion of the Gram matrix. This rules out the straight-forward use of GPs in an incremental fashion. The runtime prediction given by (2) is, however, only $O(n)$ for computing the mean prediction but $O(n^2)$ for computing the variance, assuming that the inverse Gram matrix has been stored from the training phase and that it does not change during runtime. A comprehensive reference for GPs and their use in classification and regression tasks can be found in [1].

While many instances of modifications to the GP algorithm are available that overcome this hurdle to online computation, all of these involve approximations to the GP to reduce the complexity of the representation. A number of approximation schemes are based on the observation that very few training points contribute to the bulk of learned GP. Sparse GP schemes have been proposed, among others, by Snelson and Ghahramani [2], who based their approximation on learning "active points", and Csato and Opper[3], who proposed Online GPs that learn and update a sparse set of basis vectors. A unifying synthesis of these and other sparse GP methods is given in [4].

If the number of active points is D and the number of training points is n, these methods reduce the complexity of updating the GP to $O(nD^2)$ from the initial $O(n^3)$. In contrast, OSMGPs are $O(n)$ without any approximations, albeit for the special case of compact kernels.

We now present Online Sparse Matrix Gaussian Processes that can perform exact incremental updates to the GP in $O(n)$ time for kernel functions with local support.

3 Online Sparse Matrix Gaussian Processes

The proposed OSMGP algorithm works under the assumption that the covariance matrix of the GP is sparse. The use of kernel functions having local support results in most of the entries in the Gram matrix being close to zero since the kernel decays rapidly

as the distance between the vectors being evaluated increases. Many commonly used infinite-dimensional kernels have local support, a case in the point being the widely used Radial Basis Function (RBF), also known as the Gaussian or squared exponential kernel. This allows the use of sparse matrix algorithms that can perform online updates in linear time and are also exact, in contrast to the existing sparse GP algorithms.

To ensure the sparsity of the covariance matrix, we use "compactified" kernel functions [5]. This is because although kernels such as the RBF may produce a covariance matrix with many small entries, the entries in the matrix should be exactly zero for sparse matrix algorithms to be applicable. While thresholding the entries of the covariance matrix may seem the most straight-forward way to obtain a sparse matrix, this may result in the matrix not being positive definite. Compactifying the kernel function, i.e., modifying the kernel function to get one with compact support, ensures a positive definite matrix without compromising on the other characteristics of the kernel. For example, the RBF kernel can be compactified as

$$k(\mathbf{x}_i, \mathbf{x}_j) = c \exp\left(\frac{(\mathbf{x}_i - \mathbf{x}_j)^2}{\eta^2}\right) \times \max\left(0, 1 - \left|\frac{\mathbf{x}_i - \mathbf{x}_j}{d}\right|\right)$$

where c and η are the RBF kernel parameters, and d defines the compact region over which the kernel has support. This modified kernel is positive definite [5].

The main difference between the OSMGP and existing GP algorithms is the representation of the covariance matrix as a sparse Cholesky factor. This is possible because the Cholesky factor of a sparse matrix is also sparse for some reordering of the variables. In the following discussion, we will take the upper triangular Cholesky factor to be the quantity that is maintained and updated.

3.1 GP Updates Using Kullback-Leibler Divergence

At each step during online learning of the GP, the algorithm is presented with a training pair that is used to update the existing model. The GP posterior is computed by taking into account the training output. Assuming at time t, the model is given by $p_t(f)$, it is updated upon receiving the training output y_{t+1} using Bayes law as $p_{t+1}(f) \propto p(y_{t+1}|f)p_t(f)$ where $p(y_{t+1}|f)$ is the measurement model.

Following Csato and Opper [3], we find the GP closest to the true posterior in the Kullback-Leibler divergence sense. This is done through moment matching using the parametrization lemma of [3]. Subsequently, we get the updated GP as

$$\langle f \rangle_{t+1} = \langle f \rangle_t + q^{(t+1)} \mathbf{k}_{t+1}, \quad K_{t+1} = K_t + r^{(t+1)} \mathbf{k}_{t+1} \mathbf{k}_{t+1}^T \tag{3}$$

where $\langle . \rangle$ denotes the expectation operation, $\mathbf{k}_{t+1} = [K(\mathbf{x}_{t+1}, \mathbf{x}_1), \ldots, K(\mathbf{x}_{t+1}, \mathbf{x}_t)]^T$ and the update variables q and r are given as

$$q^{(t+1)} = \frac{\partial}{\partial \langle f_{t+1} \rangle_t} \ln \langle p(\mathbf{y}_{t+1}|f_{t+1}) \rangle_t, \quad r^{(t+1)} = \frac{\partial^2}{\partial^2 \langle f_{t+1} \rangle_t} \ln \langle p(\mathbf{y}_{t+1}|f_{t+1}) \rangle_t \tag{4}$$

where $\langle . \rangle_t$ is the expectation with respect to GP at time t.

Updating the GP model using (3) involves a $O(n)$ update for the mean and an update for the covariance that is potentially $O(n^2)$. Here n is the number of training samples

presented thus far to the algorithm. However, this is a rank one update to a sparse matrix where the dimensions of the matrix increase by one during this update. The Cholesky factor of the covariance can, in this case, be updated in $O(n)$ through the use of Givens rotations which is described next.

3.2 Givens Rotations for Incremental Covariance Matrix Update

A standard approach to perform efficient, incremental Cholesky factorization uses *Givens rotations* [6] to zero out the entries below the diagonal, one at a time. To zero out the (i, j) entry, a_{ij} of a matrix A, we apply the Givens rotation

$$G \triangleq \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \tag{5}$$

to rows i and j, with $i > j$, which represents a rotation in a two-dimensional subspace of the states. ϕ is chosen so that a_{ij}, the (i, j) entry of the matrix, becomes 0.

$$(\cos\phi, \sin\phi) = \begin{cases} (1, 0) & \text{if } \beta = 0 \\ \left(\frac{-\alpha}{\beta \sqrt{1+\left(\frac{\alpha}{\beta}\right)^2}}, \frac{1}{\sqrt{1+\left(\frac{\alpha}{\beta}\right)^2}} \right) & \text{if } |\beta| > |\alpha| \\ \left(\frac{1}{\sqrt{1+\left(\frac{\beta}{\alpha}\right)^2}}, \frac{-\beta}{\alpha \sqrt{1+\left(\frac{\beta}{\alpha}\right)^2}} \right) & \text{otherwise} \end{cases}$$

where $\alpha \triangleq a_{jj}$ and $\beta \triangleq a_{ij}$. This is illustrated in Figure 1 which shows how a Givens rotation can be applied to a matrix R which is triangular but for one entry. Note that the single Givens rotation does not ensure that the resulting matrix R' is triangular since the operation may give non-zero values to other elements to the right of a_{ij} in the ith and jth rows, shown in red in Figure 1.

After all the non-zero entries below the diagonal are zeroed out by application of Givens rotations, the upper triangular entries contain the new Cholesky factor. Note that a sparse matrix yields a sparse Cholesky factor for an appropriate variable ordering.

Applying Givens rotations yields an efficient update algorithm. In general, the maximum number of Givens rotations needed for adding a new row of size n is $O(n^2)$. However, as both the covariance matrix and the new row are sparse, only $O(n)$ Givens rotations are needed. We observe that in practice it is typically much faster than this.

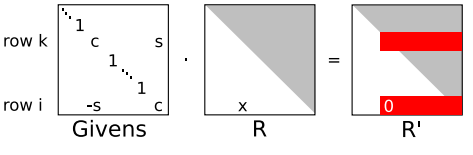


Fig. 1. Using a Givens rotation as a step in transforming a general matrix into upper triangular form. The (i, j)th entry, marked 'x' here, is eliminated, changing some of the entries in the ith and jth rows marked in red (dark), depending on sparsity.

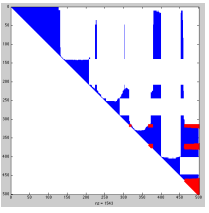


Fig. 2. Updating a sparse Cholesky factorization after adding a row and column is $O(n)$ using Givens rotations for a $n \times n$ matrix but is often much faster in practice. In the above sparsity pattern of Cholesky factor (non-zero entries are marked in blue or red), entries whose values have changed after the update are shown in red and unchanged entries are shown in blue. In this 500×500 matrix, only about 1,500 entries are modified.

As an example, Figure 2 shows the entries in a matrix of size 500×500 that change value upon updating the factorization after adding a row and a column. The algorithm would be $O(n^2)$ if the whole matrix needed to be recomputed. However, only a small number of entries are recomputed in practice.

4 Variable Reordering and Hyperparameter Optimization

While the above sections dealt only with GP update, the need to periodically learn the GP hyperparameters and maintain the sparsity of the Cholesky factor of the Gram matrix can cause inefficiencies. We address these issues in this section.

First, while the parameters can be learned offline using an initial training set, this makes the GP less responsive to changes in the test set during runtime. However, once new hyperparameters are available, the covariance matrix has to be recomputed completely and re-factorized using a batch Cholesky decomposition. This operation could take $O(n^3)$ in theory but is closer to $O(n^2)$ in practice if sparse Cholesky decomposition methods are used [7].

Second, the runtime of the OSMGP depends crucially on the sparsity of the Gram matrix. However, as Givens rotations are incrementally performed to update the GP, *fill-in* can occur in the Gram matrix. Fill-in is defined as non-zero entries beyond the sparsity pattern of the Gram matrix, i.e., entries that are zero in the Gram matrix become non-zero in the Cholesky factor. This occurs because the Cholesky factor of a sparse matrix is guaranteed to be sparse for some variable orderings but not for all of them.

We avoid fill-in by *variable reordering*, a technique well known in the linear algebra community, using a heuristic to efficiently find a good column ordering. The order of the columns (and rows) in the Gram matrix influences the variable elimination order and therefore also the resulting number of entries in the Cholesky factor. While obtaining the best column variable ordering is NP hard, efficient heuristics such as the COLAMD (*col*umn *a*pproximate *m*inimum *d*egree) ordering [8] and Nested Dissection [9] perform well in practice. The Cholesky factor after applying the COLAMD ordering, for instance, shows negligible fill-in, as can be seen in Figure 3. Reordering the variables also needs a re-factorization of the Gram matrix with its attendant higher complexity.

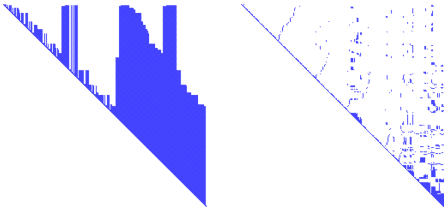


Fig. 3. The sparsity pattern of the Cholesky factor (non-zero entries marked in blue) for factorization without reordering (left), and using COLAMD reordering (right). The reordered Cholesky factor has very few non-zero entries and so is much sparser.

We propose fast incremental updates with periodic variable reordering and hyperparameter optimization using (1), where the latter two are combined in a single step, thus requiring only a single re-factorization. When combined with incremental updates, this avoids fill-in, relearns hyperparameters to provide a responsive model, and still yields a fast algorithm as is supported by the results in Section 6.

5 Matrix Downdates and $O(1)$ Operation

The complete OSMGP algorithm as described above has $O(n)$ runtime complexity. This is because the GP update step (3) has $O(n)$ runtime due to the use of Givens rotations while the regression prediction (2) also has $O(n)$ runtime since it can be implemented using sparse back-substitution. The mean prediction can be found by computing $(K + \sigma^2 I)^{-1}\mathbf{y}$ as the solution to the linear system $(R^T R)\mathbf{x} = \mathbf{y}$ where R is the upper triangular Cholesky factor of the Gram matrix. This linear system can be solved using two back-substitution operation. Although in normal operation, back-substitution is an $O(n^2)$ operation, it is $O(n)$ for sparse matrices.

While the linear runtime complexity may be good for most applications, in many other situations we require a constant time scaling, at least for the prediction. Further, since the covariance matrix in the above case grows with the number of training samples, storage requirement also increases over time. We can get around these constraints and obtain a constant time algorithm by introducing approximations into the current scheme, which is exact except for the posterior projection (4).

Our approach is based on the sliding window approach to least squares problems [10], where old measurements are successively discarded as new ones arrive. For the OSMGP, this implies discarding an old training sample for every new one that is provided, thus keeping the number of samples based on which the GP is learned constant. We propose this "oldest first" discarding strategy since, during online operation, it is more likely that future test samples are similar to the latest observed samples. However, other discarding strategies can also be accommodated in the algorithm.

Maintaining the covariance matrix at a fixed size of $W \times W$, where W is the window size, makes both the prediction and the GP updates have $O(W)$ time instead of $O(n)$. Further, even the hyperparameter optimization and variable ordering can be done in $O(W^2)$. Note that W can be quite large (in the thousands) and yet, can be done efficiently, since all the operations are carried out on sparse matrices

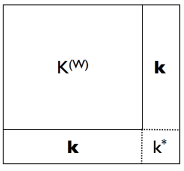


Fig. 4. Illustration of variable grouping for the downdate process. See text for explanation.

The only operation to be described in this constant time, constant space algorithm (for fixed W) is the update of the GP when discarding a training sample. Discarding a training sample involves deleting a row and a column from the covariance matrix. Inverting the rank one update from (3), this can be done using a rank one downdate of the covariance matrix. Assuming, without loss of generality, that the $(W+1)$th row and column are to be removed from a $(W+1)\times(W+1)$ matrix $\tilde{K}$ to get a downdated $W \times W$ matrix K, this can be done as

$$K = K^{(W)} - \frac{\mathbf{k}\mathbf{k}^{\mathbf{T}}}{k^{\star}}$$

where $\mathbf{k}$, $k^{\star}$, and $K^{(W)}$ are defined as in Figure 4.

The rank one downdate can be performed efficiently using Hyperbolic rotations [6], which are defined analogous to Givens rotations. To zero out the (i,j) entry, a_{ij} of a matrix A, we apply the Hyperbolic rotation

$$H \triangleq \begin{bmatrix} \cosh\phi & -\sinh\phi \\ -\sinh\phi & \cosh\phi \end{bmatrix} \tag{6}$$

to rows i and j, with $i > j$. The parameter ϕ is chosen so that a_{ij}, the (i,j) entry of the matrix, becomes 0.

$$(\cosh\phi, \sinh\phi) = \begin{cases} (1,0) & \text{if } \beta = 0 \\ \left(\frac{\alpha}{\beta \sqrt{1-\left(\frac{\alpha}{\beta}\right)^2}}, \frac{1}{\sqrt{1-\left(\frac{\alpha}{\beta}\right)^2}} \right) & \text{if } |\beta| > |\alpha| \\ \left(\frac{1}{\sqrt{1-\left(\frac{\beta}{\alpha}\right)^2}}, \frac{\beta}{\alpha \sqrt{1-\left(\frac{\beta}{\alpha}\right)^2}} \right) & \text{otherwise} \end{cases}$$

where $\alpha \triangleq a_{jj}$ and $\beta \triangleq a_{ij}$. As with the Givens rotations, we apply hyperbolic rotations until all the elements of the row and column in question have been zeroed out. This is a linear time operation for sparse matrices.

Hyperbolic rotations have a drawback that they can be numerically unstable. However, if the initial and final matrices after the downdate have full rank, which is the case here, instability usually does not occur. Though we have not encountered numerical instability in our experiments, an unstable case can be dealt with using more sophisticated downdate techniques, e.g., [11].

The approximate, constant time OSMGP algorithm can now be summarized as follows. Whenever a training sample is presented, we include it and update the GP using

Givens rotations. If the window size has not yet been reached, no more work needs to be done. However, if the window size is exceeded, we then discard the oldest training sample by removing the corresponding row and column from the Cholesky factor of the covariance matrix, using hyperbolic rotations for downdating.

6 Applications

We now illustrate OSMGPs using two challenging computer vision applications, namely head pose estimation and tracking. The accuracy of the OSMGP regression algorithms is compared against the existing methods in both cases.

6.1 Head Pose Estimation

Information about head pose provides an important cue in human interaction that people routinely estimate and use effortlessly. Our motivation for studying head pose estimation is to enable more natural interaction between humans and machines. Following a person's viewpoint provides machines with valuable contextual information and also enables faster reaction to user actions such as pointing. However, automatic estimation of head pose from visual data has proven to be a difficult problem. This is due to the difficulty in dealing with wide variations in pose angle and also with generalizing the estimation algorithm across face images of different identities. Finally, all this has to be performed in real-time to keep up with the speed of human head motion.

A number of head pose estimation techniques have been proposed , ranging from prototype matching [12] to 3D modeling [13] to feature-based tracking [14]. A few techniques that use regression also exist, e.g. using neural networks [15], and Support Vector Machines [16]. Another work related to ours is given in [17], which discusses head pose estimation using an offline semi-supervised algorithm for GP regression. However, all of these techniques are limited in that they either deal with a limited range of angles or use coarse discretization in angle space. In addition, few systems focus on real-time applicability. In contrast, our method works in a large, continuous angle space and operates in real time.

Our OSMGP-based head pose estimation system is fully automatic and incorporates face detection, tracking, and head pose estimation. Face detection is performed using a cascade detector [18], while tracking is performed using an incremental visual tracker [19]. The tracked face image is used as input to the GP regression algorithm. Histogram equalization is performed on the tracker output to remove illumination changes to some extent. Since the roll angle of the head pose is given by the orientation of the tracking window, only the yaw and pitch angles are learned using GP regression.

The tracker operates on an normalized image of size 32×32 pixels. We experiment with several dimensionality reduction techniques and opt for Principal Component Analysis (PCA) since it gives the best results in conjunction with GP regression. From our experiments, mixtures of probabilistic PCA tend to cluster training images by identity (rather than pose) for non-extreme poses, which is undesirable, while clustering face images in the subspace by the ground truth pose angles produces overlapping and incoherent clusters. Some dimensionality reduction techniques are evaluated in the results.

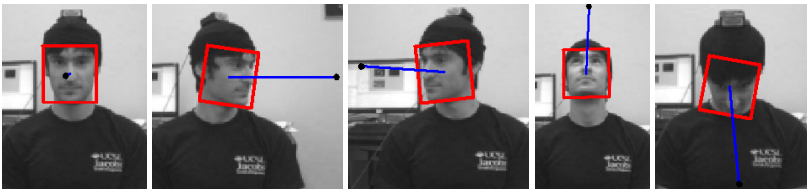


Fig. 5. Sample output of our head pose estimation system for (yaw,pitch,roll) values in degrees of (left to right) $(-2.2, -1.6, 1.8)$, $(-55.6, -6.6, 11.7)$, $(59.0, -3.3, -9.0)$, $(2.4, -24.1, 3.2)$, and $(1.0, 19.7, 11.9)$. The cap with mounted IMU is used to collect ground truth data.

We have gathered a large dataset of labeled training images. This is accomplished by instrumenting subjects with a cap on which a 3D Inertia Measurement Unit (IMU) is mounted. The IMU unit used for this purpose introduces noise into the dataset due to slippage of the IMU and due to magnetic noise in the vicinity. The data set consists of 17 subjects with over 30,000 images with ground truth pose angles. The head poses in the dataset range from -60^o to 60^o in yaw, -55^o to 40^o in pitch, and -35^o to 35^o in roll. We intend to make this dataset publicly available in the near future.

Our head pose estimation system is implemented in C++ and runs at 8 frames per second. We perform frame-based pose estimation which has the advantages of easier failure recovery and applicability to any still image. For fair comparison with other GP algorithms, we used a Matlab implementation. Publicly available Matlab implementations of the Online GP (OGP) [3] and the Sparse Pseudo-input GP (SPGP) [2] are used. In all the experiments, PCA is used to project the face images onto a 30 dimensional space on which the regression functions are learned. Our system assumes that the images are captured using a parallel frontal projection, i.e., the subject's head is orthogonal to the image plane. However, a wide variation in the head pose angles is supported. Figure 5 shows some head pose estimation results.

We first compare the efficiency of OSMGPs with OGP. For this purpose, the exact OSMGP algorithm with no downdates is used, while OGP is used with 200 basis vectors. An initial GP model is learned offline with 5,000 training points using the respective algorithms. Figure 6 shows the GP update time on a log scale for the two algorithms for a further 15,000 online updates. Note that once the maximum basis vector number is reached, OGP has constant time operation, while OSMGP has a $O(n)$ growth. However, even after 15,000 updates, the runtime of the exact OSMGP is still less than OGP. While OSMGP runtime increases monotonically if approximations using downdating are not performed, this shows that even when using an exceedingly large window size, 15,000 in this case, OSMGP is still a faster algorithm than OGP. Hyperparameter optimization is performed every 1,000 steps for OSMGPs which manifests itself as spikes in the time graph. Variable reordering is not required as the matrix remained sparse. No optimization is performed in the case of OGPs. In spite of this, the average time taken by the OSMGP per step is less than the OGP.

We compare the results of using OSMGP regression for head pose estimation against other popular kernel regression methods. The methods compared are the OGP, the Multivariate Relevance Vector Machine (MVRVM) [20], and SPGP. The publicly available implementation of MVRVM is used for fair comparison. In addition, we also evaluate

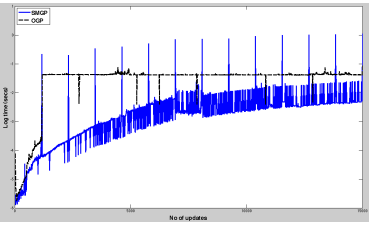


Fig. 6. Time per online update step for 15,000 steps comparing the OGP algorithm with the OSMGP. Time in seconds is shown on a log scale. Optimization is performed every 1,000th step in the OSMGP which accounts for the spikes in time.

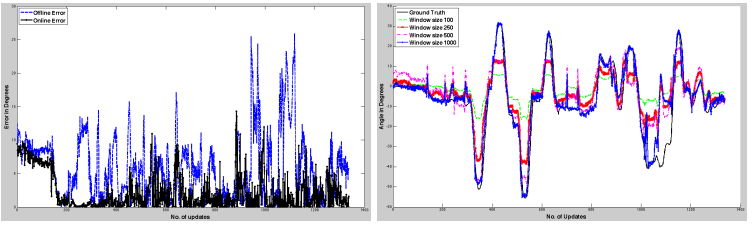
the use of different dimensionality reduction techniques in conjunction with the OSMGP, such as mixtures of PPCA with 5 clusters, mixtures of PPCA clustered by training pose with 5 clusters, and the Gaussian Process Latent Variable Model (GPLVM) [21]. OSMGP and OGP are trained online with hyperparameter estimation every 500 steps. No downdating is done for the OSMGP, while 250 basis vectors are used for the OGP. All the models are first learned offline using a 5,000 point dataset. A 15,000 point test set, the same as the one used in the timing experiment above, is used to obtain pose predictions. Note that the prediction error can be further reduced by filtering the output using the variance provided by the GP. Mean errors for the yaw and pitch angles are given in Table 1. OSMGP performs better than all the other algorithms. The OGP is slightly worse, mainly due to the approximation involved in the use of basis vectors. SPGP and MVRVM give significantly worse results due to the lack of online learning and hyperparameter estimation. Other dimensionality reduction techniques produce results that, in many cases, are rather poor. This is because the clustering is quite poor in the case of mixtures of PPCA, while GPLVM often clusters by identity of the face, rather than by pose, as desired.

A second experiment demonstrates the benefit of online learning with hyperparameter estimation. Two OSMGP models are trained offline using 5,000 training points.

Table 1. Overall pose estimation accuracy of various regression algorithms on the 15,000 point test set. Columns 2 and 3 show the error for cases when the test subject is included and excluded in the training set respectively. The first number in each entry is the mean yaw error while the second is mean pitch error. Mean roll error obtained from the tracker is 5.88^o. The last three rows tabulate the results from the OSMGP algorithm when used with different dimensional reduction techniques.

Algorithm	*Mean Pose Error 1*	*Mean Pose Error 2*
OSMGP	2.99^o, 3.07^o	6.11^o, 7.53^o
OGP	3.57^o, 3.81^o	7.13^o, 8.11^o
SPGP	4.01^o, 4.03^o	9.70^o, 10.69^o
MVRVM	4.02^o, 4.17^o	9.26^o, 10.97^o
OSMGP + Mixture of PPCA	8.80^o, 8.11^o	12.19^o, 16.64^o
OSMGP + GPLVM	10.93^o, 10.98^o	10.80^o, 11.03^o
OSMGP + MPPCA by pose	7.14^o, 8.09^o	11.58^o, 12.91^o

Subsequently, both models are used to predict poses for a sequence of 1,340 test images. The first OSMGP is updated online by providing the corresponding training point after each prediction, while the second OSMGP is kept fixed. Further, the hyperparameters of the first OSMGP are updated after every 100 steps. Variable reordering is performed in both cases as required and no downdates are applied. Figure 7(a) gives the results of the experiment in the form of the error as measured from the ground truth given by the IMU. It can be seen that although both models start by predicting poses with the same error, the model with online learning quickly improves and demonstrates much lower error over the whole sequence. The mean error over yaw and pitch for the OSMGP with online learning is 3.07^o, while for the second, fixed OSMGP, it is 7.11^o, i.e., worse by more than a factor of two.



(a) (b)

Fig. 7. (a) Comparison of mean prediction error, pitch and yaw combined, for a OSMGP with online learning as opposed to one with no online learning. (b) A comparison of predicted pitch for various window sizes over the same test sequence.

The effect of varying the window size in the approximate, constant time OSMGP is illustrated in Figure 7(b). This shows the predictive pitch angle for the same sequence of 1,340 images as above with various window sizes ranging from 100 to 1,000. No hyperparameter estimation is done in any of the OSMGPs in this case although online updates are performed. A window size of 1,000 produces a mean prediction error of 4.02 degrees in the pitch angle which is close to the value observed in the above experiment. The ground truth, as obtained from the IMU, is also shown in the figure for reference. The mean errors for window sizes of 100, 250, and 500 are 7.07^o, 6.59^o, and 5.22^o respectively.

Figure 7(b) also shows that most of the prediction error occurs in cases of extreme poses, i.e., absolute yaw or pitch angles greater than about 40^o. One reason for this is that at these angles, the visual tracker is somewhat unstable as little facial information is available. Second, the number of training images with these poses are also relatively few, since people do not maintain these positions for a significant amount of time. Correspondingly, the mean errors in Table 1 are skewed due to these extreme poses. If the pose range is constrained, prediction errors can be reduced significantly. Further, due to the skew in training data, the variance given by the GP predictor is much larger at these extreme poses than that for other poses. For instance, the maximum standard deviation of 2.59^o is obtained for a prediction of yaw and pitch values equal to 61.2^o and 31.7^o respectively. In contrast, for the corresponding angle predictions of 0.9^o and 1.2^o, the standard deviation has the more typical value of 0.28^o.

6.2 Visual Tracking

We next illustrate the application of OSMGPs to visual tracking in a regression-based framework. The tracking algorithm is similar to [22] in which an RVM is used, whereas in this work the proposed OSMGP algorithm is used to learn a regression function. As in [22], we use "seed images", where the location of the object of interest is known. The object extracted from these seed images is perturbed along the two translation axes to obtain training images using which a regression function from the image to the displacement is learned. Perturbed training images are generated in a 40 pixel window along both translation axes. Since our main goal is to demonstrate the applicability of OSMGPs, we do not take into account rotation and scaling in our tracker although these can be accommodated. We also do not perform any state prediction but simply update the tracked region on a frame-by-frame basis.

The regression-based tracker is implemented using OSMGP, MVRVM, and SPGP. Two publicly available datasets from [19], namely the Fish and Sylvester datasets are used to evaluate these algorithms. Both datasets are challenging as the target objects undergo large appearance variation due to change in illumination and viewpoints. To create the training sets, three seed images are chosen randomly in each dataset and perturbed as explained above. Ground truth is collected for computing the tracking errors of each tracker and for initialization. The tracking error of each frame is computed as $L2$ distance between the center of the tracked region and the ground truth. Further, if the tracking error of a tracker is larger than 40 pixels along any axis, the tracker is declared to have lost track and is reinitialized using the ground truth of that frame. The OSMGP with downdating and a window size of 500 is used. Both the MVRVM and the SPGP are learned with an active point set of size 100, as beyond this performance improvement is negligible. The tracking results from these experiments are presented in Table 2. The OSMGP based tracker has the least number of tracking failures and the least mean tracking error, especially for the Fish sequence. As the target object in the Fish sequence undergoes large and continuous illumination change, it is important to learn regression function in an online manner. The tracking results of the OSMGP based tracker for the two sequences is illustrated in Figure 8, and more results can be found in the supplementary material. All the trackers run at approximately 5 frames per second with MATLAB implementations on a 2.4GHz machine. The OSMGP is not significantly faster since the Gram matrix is not as sparse as in the head pose estimation scenario. As the perturbed images look alike in this type of tracking algorithm, the corresponding Gram matrix entries are non-zero and thus the matrix is relatively non-sparse. Consequently, their runtime goes up as the complexity of OSMGP depends on

Table 2. Tracking performance of different regression algorithms on the two test sequences

Algorithm	Fish Sequence		Silvester Sequence	
	# of Failures	***Mean Tracking Error***	***# of Failures***	***Mean Tracking Error***
OSMGP	6	8.3 pixels	1	1.6 pixels
SPGP	12	13.6 pixels	4	3.1 pixels
MVRVM	11	11.2 pixels	2	3.1 pixels

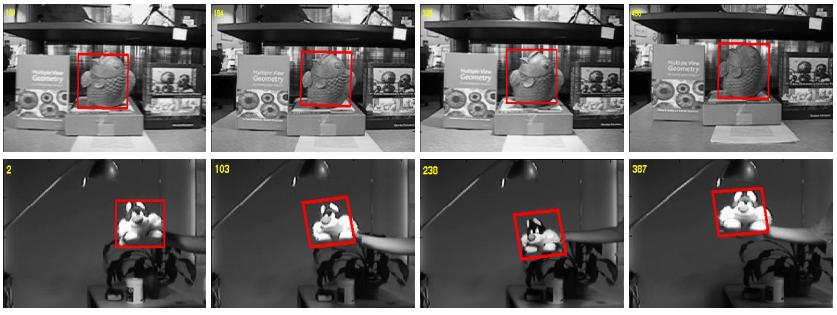


Fig. 8. Results of the OSMGP tracker for both test datasets on approximately 500 frames of video

the number of Givens and Hyperbolic rotations. In the head pose estimation case, most of the Gram matrix entries are zero as those images are obtained from different people with large pose variation. These experiments show that even in the case where the Gram matrix is not sparse, OSMGPs are still faster than the existing regression algorithms and can be significantly faster in many cases (such as the head pose estimation application).

7 Conclusion

We have presented a new Gaussian Process algorithm, Online Sparse Matrix Gaussian Processes, that can be used for exact, online learning with $O(n)$ time growth or in an approximate manner with $O(1)$ time. The algorithm is applicable to kernels with compact support, which result in a sparse covariance matrix. We demonstrated the use of the OSMGP algorithm in the context of two vision applications:

- a 3D head pose estimation system that operates in continuous angle space without any discretization, generalizes across faces of different identities, and provides good results even for large angle variations.
- a regression-based tracker that can operate under significant illumination and viewpoint changes.

In principle, the idea of sparse matrix manipulations can be extended to other probabilistic kernel machines, such as Relevance Vector Machines. We plan to pursue this idea, and extend the proposed algorithm to the semi-supervised learning setting.

References

1. Rasmussen, C.E., Williams, C.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)
2. Snelson, E., Ghahramani, Z.: Sparse Gaussian processes using pseudo-inputs. In: Advances in Neural Information Processing Systems, pp. 1259–1266 (2006)
3. Csato, L., Opper, M.: Sparse online gaussian processes. Neural Computation 14(2), 641–669 (2002)

4. Quinonero-Candela, J., Rasmussen, C., Williams, C.: Approximation methods for gaussian process regression. In: Large-Scale Kernel Machines, pp. 203–224. MIT Press, Cambridge (2007)
5. Hamers, B., Suykens, J., Moor, B.D.: Compactly Supported RBF Kernels for Sparsifying the Gram Matrix in LS-SVM Regression Models. In: Proceedings of the International Conference on Artificial Neural Networks, pp. 720–726 (2002)
6. Golub, G., Loan, C.V.: Matrix Computations. Johns Hopkins University Press (1996)
7. Kaess, M., Ranganathan, A., Dellaert, F.: Fast incremental square root information smoothing. In: Proceedings of International Joint Conference on Artificial Intelligence, pp. 2129–2134 (2007)
8. Davis, T., Gilbert, J., Larimore, S., Ng, E.: A column approximate minimum degree ordering algorithm. ACM Transactions on Mathematical Software 30(3), 353–376 (2004)
9. Kernighan, B., Lin, S.: An efficient heuristic procedure for partitioning graphs. The Bell System Technical Journal 49(2), 291–307 (1970)
10. Zhao, K., Fuyun, L., Lev-Ari, H., Proakis, J.: Sliding window order-recursive least-squares algorithms. IEEE Transactions on Acoustics, Speech, and Signal Processing 42(8), 1961–1972 (1994)
11. Bjorck, A., Park, H., Elden, L.: Accurate downdating of least-squares solutions. SIAM Journal on Matrix Analysis and Applications 15(2), 549–568 (1994)
12. Kruger, N., Potzsch, M., von der Malsburg, C.: Determination of face position and pose with a learned representation based on labeled graphs. Image and Vision Computing 15(8), 665–673 (1997)
13. Yang, R., Zhang, Z.: Model-based head pose tracking with stereo vision. In: Proceedings of the International Conference on Automatic Face and Gesture Recognition, pp. 242–247 (2002)
14. Yao, P., Evans, G., Calway, A.: Using affine correspondence to estimate 3D facial pose. In: Proceedings of IEEE International Conference on Image Processing, pp. 919–922 (2001)
15. Rae, R., Ritter, H.: Recognition of human head orientation based on artificial neural networks. IEEE Transactions on Neural Networks 9(2), 257–265 (1998)
16. Li, Y., Gong, S., Liddell, H.: Support vector regression and classification based multi-view face detection and recognition. In: Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition, pp. 300–305 (2000)
17. Williams, O., Blake, A., Cipolla, R.: Sparse and semi-supervised visual mapping with the S3GP. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 230–237 (2006)
18. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 511–518 (2001)
19. Ross, D., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. International Journal of Computer Vision 1–3, 125–141 (2008)
20. Thayananthan, A., Navaratnam, R., Stenger, B., Torr, P., Cipolla, R.: Multivariate relevance vector machines for tracking. In: Proceedings of European Conference on Computer Vision, vol. 3, pp. 124–138 (2006)
21. Lawrence, N.: Gaussian process latent variable models for visualization of high dimensional data. In: Advances in Neural Information Processing Systems, pp. 329–336 (2004)
22. Williams, O., Blake, A., Cipolla, R.: Sparse bayesian regression for efficient visual tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(8), 1292–1304 (2005)

Multi-stage Contour Based Detection of Deformable Objects

Saiprasad Ravishankar, Arpit Jain, and Anurag Mittal

Indian Institute of Technology Madras, India
sairavi45@gmail.com, arpitjain1@gmail.com,
amittal@cse.iitm.ernet.in

Abstract. We present an efficient multi stage approach to detection of deformable objects in real, cluttered images given a single or few hand drawn examples as models. The method handles deformations of the object by first breaking the given model into segments at high curvature points. We allow bending at these points as it has been studied that deformation typically happens at high curvature points. The broken segments are then scaled, rotated, deformed and searched independently in the gradient image. Point maps are generated for each segment that represent the locations of the matches for that segment. We then group k points from the point maps of k adjacent segments using a cost function that takes into account local scale variations as well as inter-segment orientations. These matched groups yield plausible locations for the objects. In the fine matching stage, the entire object contour in the localized regions is built from the k-segment groups and given a comprehensive score in a method that uses dynamic programming. An evaluation of our algorithm on a standard dataset yielded results that are better than published work on the same dataset. At the same time, we also evaluate our algorithm on additional images with considerable object deformations to verify the robustness of our method.

1 Introduction

Object detection using edge information is an important problem in Computer Vision that has received considerable attention from many researchers. The popularity of such methods is due to the fact that edges encode the object shape and are fairly invariant to color and illumination changes.

In this paper, we present a computationally efficient and robust multi-step approach for localizing objects in real, cluttered scenes given a single or few object sketches. Simple hand drawn models input by the user are first obtained for different object classes. The model is broken into segments at points of high curvatures. In the coarse search stage, these segments are passed through a series of deformations such as rotation, scale and bend and searched over the gradient image. Point maps are obtained for each segment which represent the possible locations of the matches for that segment.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 483–496, 2008.

We then connect k points from the point maps of k adjacent segments (forming a k-segment group) using a cost function that takes into account local scale variations as well as inter-segment orientations.

Each matched k-segment group is used to vote for local object centroids. Since a correct object is likely to be identified by several k-groups, maxima in centroid densities (centroid boosting) are used to obtain bounding boxes that represent likely locations of objects in the image. The contour through each matched k-segment group is built from the gradient image using bending energy and edge strength formulations. In the fine stage of matching, the contours of the whole object are matched in a sequence using dynamic programming and a score is given for the matches in order to differentiate between correct detections and false ones.

We work on the gradient image which is rich in edge information and hence, as opposed to many other methods proposed in the literature, our approach does not encounter the problems associated with the use of edge detectors such as Canny [1] or the berkeley edge detector [2] that try to reduce the clutter in images by using spatial gradient information but unfortunately also miss out on many important edges. Dynamic programming on this gradient image allows us to be efficient while searching for the object segments.

1.1 Related Work

Several approaches which use edge information for object detection have been proposed in the past. Early methods include the Hausdorff distance [3] and the Chamfer distance [4] measures between an edge-based object model and the edge image. Such methods do not allow for much change in the shape of the model, but may be used for matching different parts separately for deformable objects.

Ferrari et al. [5] have shown that objects can be detected accurately in images using simple model sketches. They build a contour segment network and find paths that resemble the model chains. The method relies on the berkeley edge detector [2] which is able to remove a lot of clutter from the edge map and detects mostly object boundaries. This method of using contour segments has been further used in [6] to group contour segments into groups of k straight segments (called kAS) which are then matched in the berkeley edge image to detect objects.

In further work [7], Ferrari et al. learn the shape model automatically from a set of training images and use a combination of Hough-style voting with a non-rigid point matching algorithm (thin-plate splines) in order to localize the model in cluttered images. The thin-plate spline model allows for a global affine transformation of the shape while allowing some local deviations from the affine model for each straight contour segment (PAS).

Opelt et al. [8] use similar ideas of combining boundary segments but learn discriminative combinations of boundary fragments (weak detectors) to form a strong Boundary-Fragment-Model (BFM) detector. Their segments encode information about position relative to the centroid and they use a hough-style voting technique to find the centroids of the objects. [9] also use probabilistic

extension of the Generalized Hough Transform to determine the object's centroid during recognition. Shotton et al. [10,11] learn discriminative contour fragments but look at these segments in isolation rather than in groups. Their fragments, however, are much larger than those of others. Similarly, Wu and Nevatia [12] use edgelet features in order to build a classifier for detection and segmentation of specific objects.

Wu et al. [13] use Gabor wavelet elements at different locations and orientations as deformable templates to learn an active basis from a set of images. Each of these elements is allowed a small amount of variation in their location and orientation while matching.

While the above mentioned methods are effective for many kinds of shape deformations, all of these methods measure deviation of each segment with respect to the object centroid or a global shape and can thus handle only relatively small local deformations of the shape from an overall shape. In contrast, we develop an approach where we use cost functions that take into account the deformations/orientations of adjacent segments. Thus, we can handle larger deformations of the object while still maintaining the overall sense of the object shape.

Felzenszwalb and Schwartz [14] use shape trees that use a hierarchical structure such that every segment is divided at different levels in a tree structure. Noise is added to each node in the shape tree and this yields the set of possible deformations of an object. An efficient algorithm based on dynamic programming was used. While this is an interesting approach, it is not clear whether such shape trees capture all possible deformations of an object.

Basri et al. [15] proposed interesting analytical functions which can be used to match deformable shapes but the results were shown only on segmented objects. We borrow many of the ideas for shape deformation from this paper, while applying them in a more general and difficult setting.

In the rest of the paper, sections 2 and 3 describe the coarse and fine stages of our object recognition algorithm while section 4 summarizes the results.

2 Coarse Match

2.1 Basic Segment Match

The first step in our approach involves breaking simple hand drawn models into segments. A single model is usually sufficient for most objects (eg. bottle, applelogo etc.) but if the object appears drastically different from some view points, a model can be obtained for each of those view points (eg. side and rear views of a car). The model is broken at points of high curvatures (sharp turns or bends) into segments of low curvature. The breaking of the model into low curvature segments is done to allow more bending deformations at the points of high curvature (similar to [15]). Figure 1 shows the segment breakup for a model.

We permit bending deformation at the points of high curvature by allowing the two low curvature segments on either side of the high curvature point to rotate with respect to each other. Each segment is allowed rotation in steps in

the range $[-\delta, \delta]$ (we use a δ of 30 degrees) to account for bending deformation at the high curvature points. Segments are also scaled to sizes in a range: $\alpha, 2\alpha, 3\alpha, ..., p\alpha$ where p is the size of the image and α is a fraction. Model segments which have small curvature typically deform the least. We allow for small bending deformations of these model segments by scaling them along the direction perpendicular to their curvature (Figure 1 (a)).

Line-like segments are easier to match but give numerous matches whereas slightly curved segments impart some degree of discriminability in the matching stage. Each rotated, scaled and deformed model segment is independently matched in the gradient image by finding paths that satisfy a normalized edge score (N).

$$N = \sum_{i,j \in C} G(i,j)/\sigma_C \tag{1}$$

where C represents the path, G is the gradient magnitude image and σ_C is the variance of the gradient values in the bounding box of the path. The maxima of N for each model segment for all its allowed scales, rotations and deformations are obtained. These can be computed quite efficiently using a sliding window mechanism that is implemented using dynamic programming.

These maxima (we take the top 15%) are represented by the mid points of the particular matched paths. Thus, the basic match step gives a point map for every segment that indicates the strong presence of that segment at those locations. The margins for bending deformations (Figure 1 (a)) and scaling of the model segments allow us to capture most of the candidate match locations. Object segments in the image which could still not be correctly matched at this stage due to their larger deformation are efficiently detected in our final detection stage.

2.2 *k*-Segment Grouping

In the basic segment match step, point maps indicating matched segment locations are generated for every model segment. The point maps encode the midpoints of these matched segments. We search for k-segment groups in the point maps to localize the objects accurately. k matched points corresponding to k adjacent segments are searched jointly and costs are enforced on these k-segment groups to obtain those that might belong to the object. The k-segment groups are obtained for all possible k adjacent segment combinations of the model.

At high values of k, we get a higher order shape perspective but the method becomes vulnerable to occlusions while at lower k values, we get a large number of matches which increases the computational cost. Hence, we choose an intermediate value of $k = 3$.

All the model segments are numbered such that adjacent segments get consecutive numbering. Adjacent segments have local scale and orientation information and hence can be used efficiently in the search process. The model segment midpoints are chosen to form the model k-segment groups.

A particular k-segment group obtained from the image point maps (the k^{th} point is extracted from the point map of the k^{th} segment in the group) is required

(a) (b) (c)

Fig. 1. Illustration of model segments extraction: (a) Bending deformation through anisotropic stretching (b) The Applelogo model from the ETHZ dataset [5] (c) The various segment end points are indicated as green dots

to satisfy local scale and orientation constraints. The k points are first linked pairwise (1-2, 2-3,.....,$(k-1)$-k) using line segments. This produces a graph (G) with k nodes, $k-1$ adjoining line segments and $k-2$ node angles. A corresponding graph is also obtained for the model. Figure 2 (a) illustrates this procedure for a model with $k = 3$.

Cost function: The scales of the line segments (obtained by joining adjacent pair of points) in the graph G with respect to the corresponding line segments in the model graph are obtained. The scale ratio $\sigma_{m,t}$ of a particular linking line segment is the ratio of the lengths of the line segment in the model (m) and the image (t): $\sigma_{m,t} = \frac{d_t}{d_m}$. The scale cannot change drastically from one line segment to the next adjoining line segment (i.e. local scale variations should be small). But far away segments can have dissimilar scale variations. The node angles (represented by α) also indicate the inter-segment structural information. A cost function for the k-segment groups involving both local scale changes and node angle changes is formulated as follows:

$$C(s, a) = w_s C_s(m, t, k) + w_a C_a(m, t, k) \tag{2}$$

where s represents scale and a the node angle. We empirically determined $w_s = 0.4$ and $w_a = 0.6$). Furthermore, we compute the cost of scale changes as

$$C_s(m, t, k) = \frac{1}{k-2} \sum_{i=1}^{k-2} (max(\frac{\sigma^i_{m,t}}{\sigma^{i+1}_{m,t}}, \frac{\sigma^{i+1}_{m,t}}{\sigma^i_{m,t}}) - 1) \tag{3}$$

where $\sigma^i_{m,t} = \frac{d_{i,i+1,t}}{d_{i,i+1,m}}$ is the scale ratio of the i^{th} segment. Cost of orientation changes is taken as:

$$C_a(m, t, k) = \frac{1}{k-2} \sum_{i=1}^{k-2} (1 - e^{-c(\frac{\Delta\alpha^i_{m,t}}{\pi})^2}) \tag{4}$$

where $\Delta\alpha^i_{m,t} = \alpha^i_m - \alpha^i_t$ is the change in the i^{th} node angle and this angle difference is normalized by π. c is a factor that controls the amount of bending

allowed by penalizing the change in the node angle. The scale and angle scores are averaged over the group. We use a threshold $C <= 0.25$ in our implementation for considering the k-segment group as matched.

Centroid Boost: Each point of a matched k-segment group contributes to a local object centroid (Figure 2 (d)). The centroid is obtained from a matched group point (p) as follows: The vector joining the corresponding model group point to the model center is scaled by the local scale calculated at p and is used to determine the centroid. Each point of a matched group contributes a candidate centroid.

Matched k-segment groups of the object in the image would provide a high centroid density about the object centroid. The centroids which are not contributed by object groups are usually scattered. This leads to centroid density boosting at correct object locations which is used to localize objects. A centroid density map (ρ) is generated for the image. Points of maxima in this map are obtained by a summation over a circular window W:

$$F = \sum_{i,j \in W} \rho(x - i, y - j) \tag{5}$$

where (x, y) is a point on the centroid density map and the summation over the circular window W allows for uncertainty about the exact centroid. The points in the map which maximize F are taken as possible object locations. Once the centroids are identified, bounding boxes are computed for each of them. For computing the scale of a bounding box with respect to the model bounding box, the average (μ) and standard deviation (σ) of the scales of the k-segment groups which contribute to centroids in the window W (about the strong centroid) are considered. The scale of the box is taken as $\mu + 2\sigma$. The matched k-segment groups lying inside the bounding boxes are considered for the fine matching stage.

In our coarse stage, we also match single stepped segment groups (eg. $1, 3, 5$) along with the adjacent segment groups as single stepped segments impart a higher order shape perspective. This is also useful for certain classes of objects where some adjacent segments are more deformable (eg. variations in the shape of a bottle) than single stepped segments.

The principle of centroid density boosting can efficiently localize multiple object instances in an image. The coarse segment group match accounts for local bending deformations in the object but does not take care of rotations of the object. To account for object rotation, we rotate our model and perform the coarse match. Rotated objects will provide maximal centroid density at the corresponding model rotation. The coarse matching stage effectively reduces the search space for the object contour detection to a few bounding boxes thus making our method computationally efficient. The effect of clutter is also drastically reduced in the final object contour search.

3 Fine Matching and Contour Completion

The coarse matching stage localizes objects with bounding boxes. The object contours are obtained in these localized regions by searching for contours in the gradient image that connect the points of the matched k-segment groups.

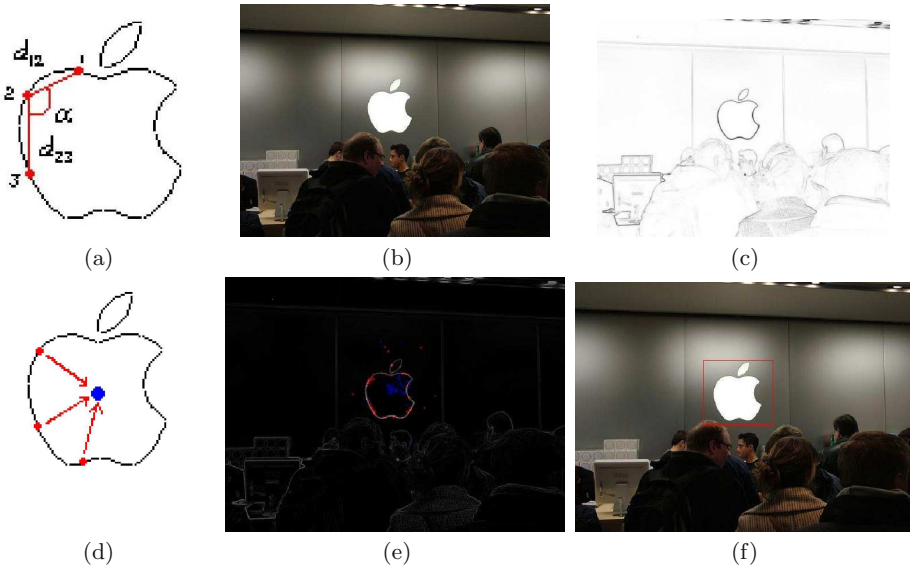


Fig. 2. Illustration of the steps of our object recognition algorithm. (a) Applelogo model with one 3-segment group (linking line segments and node angle α) shown (b) A test image from the database (c) The gradient magnitude image (d) Example of Centroid boosting (e) Matched k-segment groups in red with the centroids in blue (f) Object localization with bounding box.

Contour Search in Oriented Boxes: The object contour is built starting with adjacent pairs of points in the matched k-segment groups (i.e. 1-2, 2-3,....,$(k-1)$-k) and completing the contour between them. An oriented box is placed between every point pair (Figure 3). The orientation of the box is given by the orientation of the line segment in the model that joins the two points. The width of the box depends on the local scale (obtained from the coarse match) and the amount of bending of the contour that can be tolerated.

The contour of a k-segment group is built progressively from point 1 to point k using oriented boxes which greatly simplifies the computational complexity of obtaining matched contours. Paths are taken between the two points in each oriented box based on continuity of gradient magnitudes and directions. A cost is formulated to obtain the best matching path that takes into account the bending energy and edge strength of the path.

The cost function for a contour path is as follows:

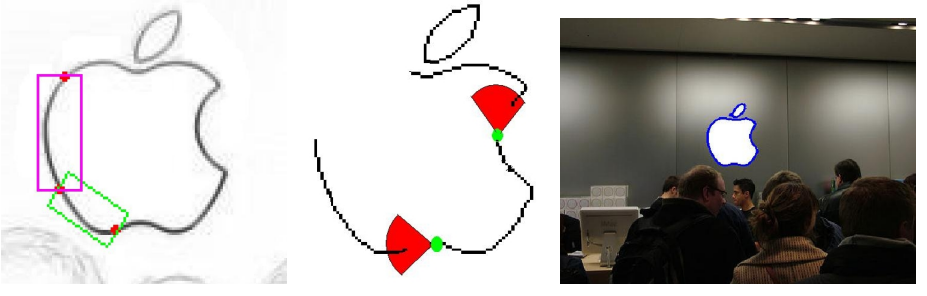
$$q = w_b C_b + w_g C_g \tag{6}$$

where C_b and C_g are the costs for the bending energy and edge strength respectively (we empirically determined $w_b = 0.7$ and $w_g = 0.3$). The bending energy cost depends on the angles between consecutive tangents along the path. A vector (ϕ_c) of these angles is obtained for the contour path in the image. A

similar vector (ϕ_m) of the angles between consecutive tangents along the corresponding segment of the model is also obtained. The larger of the two vectors is downsampled and the bending energy cost is computed as follows:

$$C_b = \frac{1}{L}\sum_{i=1}^{L}(1 - e^{-}(\frac{\phi_c^i - \phi_m^i}{\pi})^2) \tag{7}$$

where L is the minimum of the vector lengths of ϕ_m and ϕ_c. The difference $\phi_c^i - \phi_m^i$ (the difference in the i^{th} elements of ϕ_c and ϕ_m) in the cost function is normalized by π. The edge strength cost C_g is obtained from the gradient magnitude image in a similar manner as the normalized edge score (Eq. 1) that was used in the basic segment match stage. The path between the two points in the oriented box which minimizes the cost q is taken as the matched contour. This contour extraction procedure is iterated for all adjacent pairs of points in the matched k-segment groups.



a b c

Fig. 3. (a) The oriented boxes for a particular 3 segment group are shown on the gradient image of the example used in Figure 2. (b) An example of the neighborhood search for contour completion using oriented sectors (c) Final object contour detection result shown in blue for the example used in Figure 2.

Fine Match using Neighborhood Search: The contours of k-segment groups (k-segment contours) obtained from the gradient image are now stitched together to obtain the full object contour in the localized boxes. The entire contour is built in steps and a cost function is updated at each step. The following are the steps of our fine object contour detection and completion algorithm:

1. The best k-segment contour (best in terms of the object contour cost discussed below) is used to start the contour build up in each of the localized bounding boxes.
2. The neighborhood at both ends of the best contour is searched to obtain the next k-segment contour candidates (refer to Figure 3 (b)).
3. The contour in the gradient image that connects the two k-segment contours is obtained. A comprehensive object contour cost (discussed below) is assigned to this extended contour. In the case of multiple candidates for contour extension, the one that gives the least total cost is chosen.

Steps 2 and 3 are iterated (with the extended matched contour updated as the best contour at step 2) till either the entire object contour is completed or the local neighborhood at both ends of the contour has no more candidate segment groups that satisfy the cost. The object contour completion algorithm is also started from a few other locations other than the best contour to obtain contours missed out in the first search.

The search for candidate segment groups looking from both ends of the current contour is implemented as follows: A sector with its origin placed at the end point of the contour and radius determined by the maximum search distance is used (Figure 3). The orientation of the sector is along the tangent direction at the endpoint of the contour. We use a sector of radius 6 pixels and angle of 60^o for an image of size 357×216 pixels.

Object Contour Cost: The cost function for the extended contour at each iteration of the algorithm is computed as follows:

$$Q = w_s C_s(m,t) + w_a C_a(m,t) + w_b C_b(m,t) + w_g C_g(m,t) \tag{8}$$

where C_s is the scale cost (Eq. 3) , C_a the angle cost (Eq. 4), C_b the bending energy cost (Eq. 7) and C_g the edge strength cost of the updated contour (Eq. 1). We empirically determined $w_s = 0.2$, $w_a = 0.2$, $w_b = 0.5$ and $w_g = 0.1$.

The first two components of the cost function Q constrain the local scales and node angles of the extended segment group (obtained by combining the current and candidate k-segment groups). They help account for inter-segment scale and orientation variations. The last two components of Q are those associated with the extended contour (bending energy and edge strength) and account for intra-segment bending deformations.

All the components of Q have already been pre-computed except at the links between the two contours (obtained at step 3 of the algorithm). We use dynamic programming to efficiently update the comprehensive contour cost at each iteration of the algorithm.

The cost at the end of each iteration is compared with a threshold to determine if we should continue matching. All the matched paths are retained in the next iteration. This type of dynamic programming helps in finding the best contour in a given bounding box efficiently and accurately. The full object contour in each bounding box is progressively built. Wrong contours or matched clutter falling inside the bounding boxes will be discarded by the cost function. Thus, the fine matching stage gives the object contours accurately.

4 Results of Experiments and Discussion

We tested our algorithm on the ETHZ database [5] which has 5 object classes namely Applelogo, Swan, Bottle, Mug and Giraffe. It contains a total of 255 images divided among object classes as apple logos (40), bottles (48), giraffes(87), mugs (48) and swans (32). The dataset contains objects of various scales, orientations, deformations and images with multiple object instances which makes it

highly challenging for object detection. We use the simple hand drawn models of [5] for our experiments. We also evaluated the algorithm on 50 additional images obtained from Flickr, each having instances of one of the object classes with considerable shape variations and clutter.

Figure 4 and 5 show some of our results for the database. Images 1-c, 2-c, 4-c, 4-d and 5-g are from the additional dataset while the rest are from the ETHZ database. The results of object contour detection are shown as white bordered lines in the image. Images 4-c, 4-e, 6-h and 7-k show the detection of object contours in very cluttered scenes by our algorithm. The object contour forms only a fraction of the scene in these cases. Images 3-e, 5-k, 7-h and 7-i show our detection results for images with multiple object instances. Images 3-e and 5-k also show detection results of multiple objects across scales present in the same image. In image 5-k, two swans of drastically different sizes are detected by our algorithm. Two of the other swans in image 5-k are near reflections of the model and hence require a reflection of the model to be detected.

Images 2-f, 4-b and 5-i show our detection of applelogos which are heavily deformed compared to the model. Image 4-b is obtained by applying an affine transformation to image 3-b. The results for swans (2-b, 2-d, 1-e, 2-e, 5-k, 6-i) also indicate that our method can handle intra-class variability very well. Image 6-j shows our detection result for a substantially rotated applelogo. We can efficiently handle multiple object instances and object rotation using the centroid boosting principle. Our algorithm successfully extracts the contour (silhouette) of giraffe in test images where there is substantial pose change (1-b, 1-c, 1-f, 3-a, 3-c, 4-a, 5-j, 7-j). Image 4-d shows an example where our algorithm correctly discriminates between an apple and an applelogo based on curvature.

A detection is counted as correct if its bounding-box overlaps more than 50% with the ground-truth one. Our system achieves a very promising average detection rate of 91% for the entire dataset of 305 images at a low value of 0.2 FPPI (False Positives Per Image obtained over all the images). In contrast to [5], we obtain the object bounding box very accurately using the mean and standard deviation of local scales contributing to the object centroid.

Table 1 shows the comparison of results of Ferrari et. al [7] and Ferrari [5] against our method for the ETHZ database alone. The detection rates at 0.4 FPPI and 0.3 FPPI averaged over the entire ETHZ dataset are shown in the table. The results illustrate that our algorithm performs remarkably well on all the object classes. Huge improvements with respect to [5] and [7] are seen for the giraffe and applelogo classes since we efficiently account for object deformation. Our method is more robust to false positives. The false positives are systematically filtered out at each step of our multi stage approach. Our cost functions are also more comprehensive compared to [5] and [7] since they take into account various factors such as bending energy, intra segment orientations, local scales and edge strength. The computational cost is also quite low (9-10 seconds on an average). We also measure how accurately the output shapes differ from the true object boundaries in a manner similar to [7]. Our method performs quite well in this respect achieving an average error rate of about 2% on the ETHZ dataset.

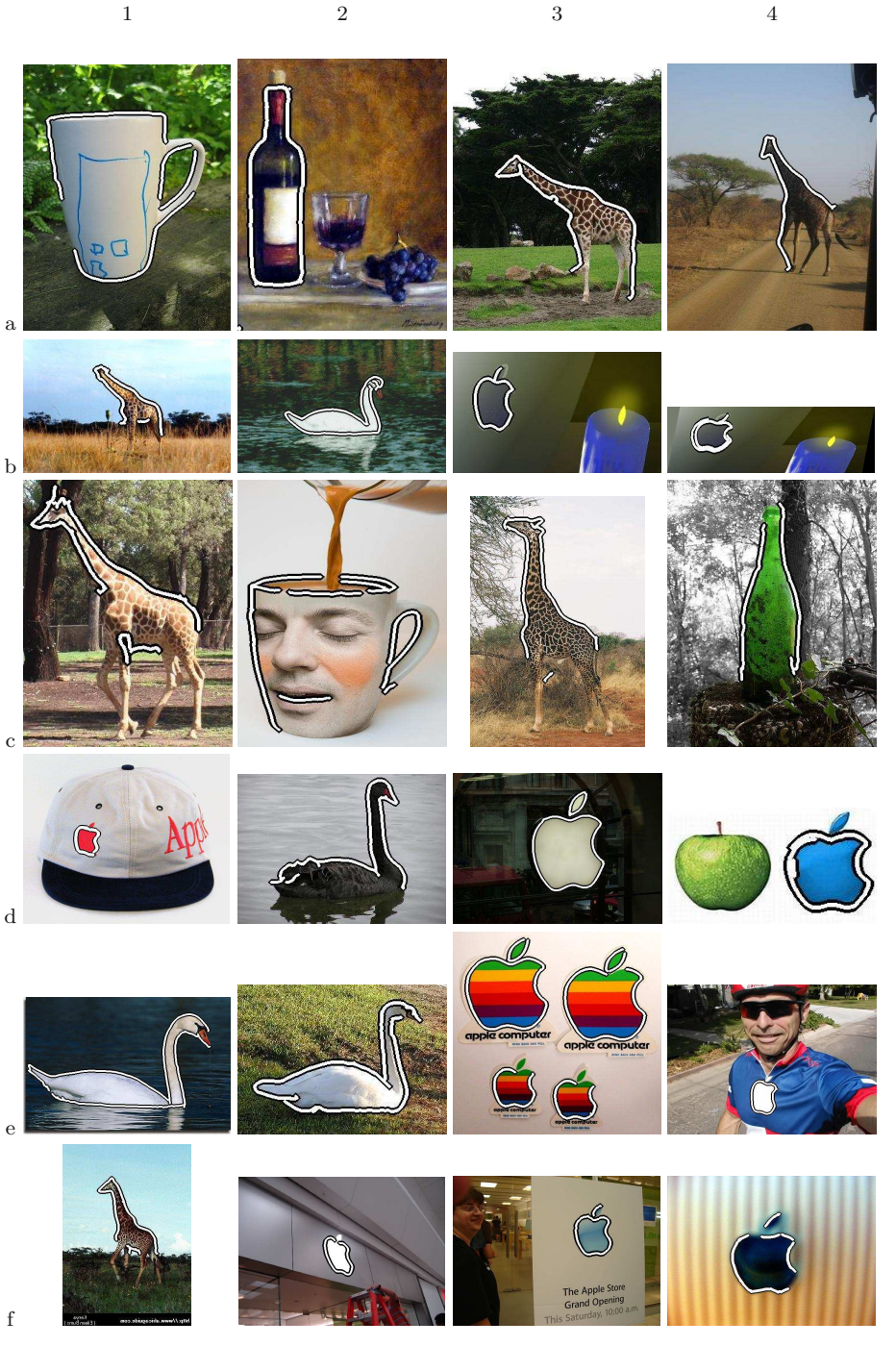

1
2
3
4
a
b
c
d
e
f
apple computer
apple computer
The Apple Store
Grand Opening
This Saturday, 10:00 a.m.

Fig. 4.

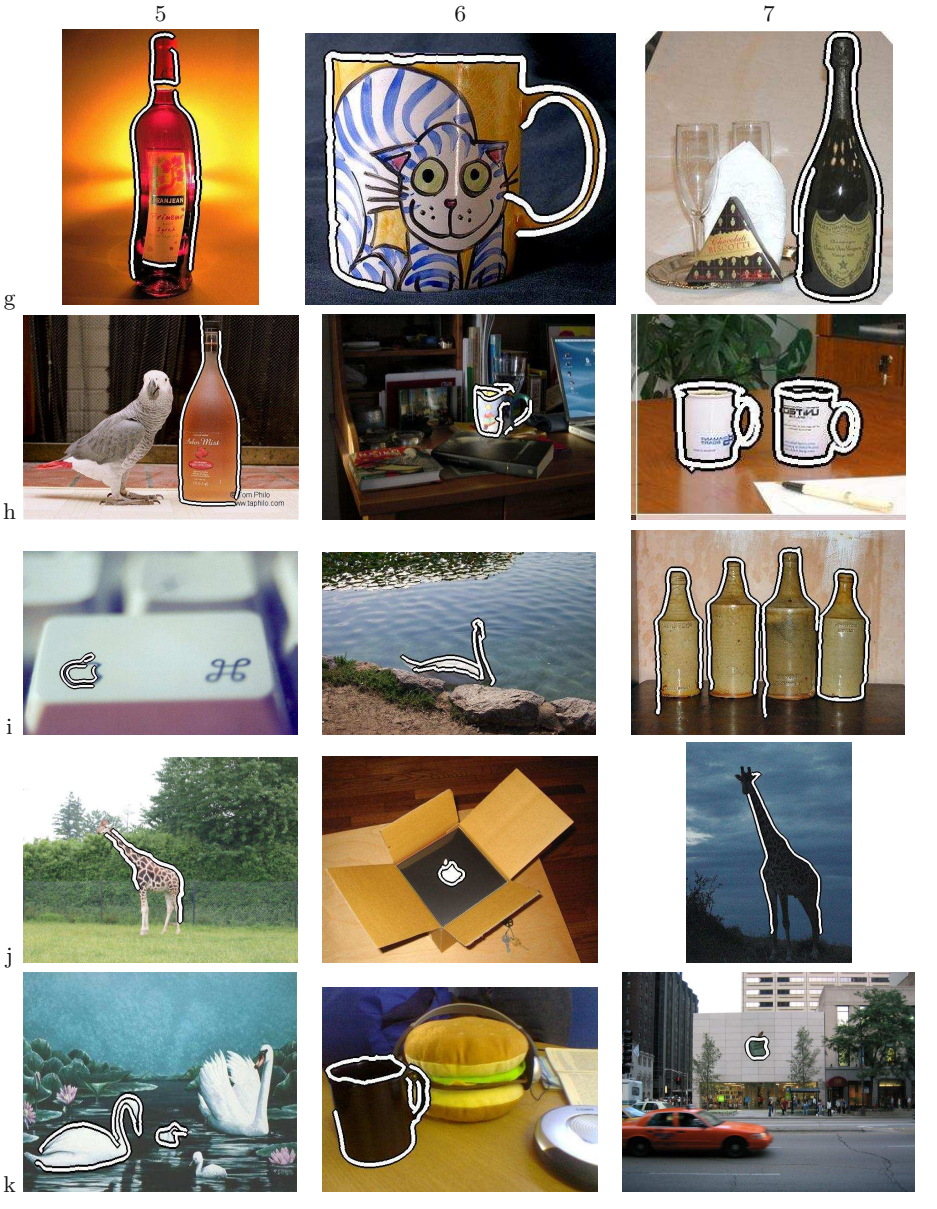


Fig. 5.

Table 1. Comparison of detection rates of objects at 0.4 FPPI / 0.3 FPPI and accuracy at 0.4 FPPI

	Apple	Bottle	Giraffe	Mug	Swan
Ferrari et al. [5]:	72.7/56.8	90.9/89.1	68.1/62.6	81.8/68.2	93.9/75.8
Ferrari et al. [7]:	86.4/84.1	92.7/90.9	70.3/65.9	83.4/80.3	93.9/90.9
our system :	97.7/95.5	92.7/90.9	93.4/91.2	95.3/93.7	96.9/93.9
accuracy:	1.2	2.1	2.3	2.9	1.8

5 Conclusion

In this paper, an efficient multi-stage approach to object recognition in real, cluttered images that is robust to scale, rotation and intra class variability is presented. Shape information from simple model sketches is used to localize objects and detect their contours. Experiments confirm that k-segment grouping together with centroid boosting can localize the objects accurately in an image. Finally the object contours are extracted in the fine matching stage using a comprehensive score and dynamic programming. Separation of the matching into two stages allows us to detect objects fast while maintaining accuracy and matching of only k-segment groups initially allows to detect objects that may be partially occluded or cluttered. Thus, the method handles local scale variations, bending deformations, clutter, multiple object instances and rotations of the object in an efficient manner and achieves results that are quite promising. Future work would involve combining other object properties like color and texture along with edge information to detect objects.

References

1. Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8, 679–698 (1986)
2. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 530–549 (2004)
3. Huttenlocher, D., Klanderman, G., Rucklidge, W.: Comparing images using the hausdorff distance. IEEE Transactions on Pattern Analysis and Machine Intelligence 15, 850–863 (1993)
4. Thayananthan, A., Stenger, B., Torr, P., Cipolla, R.: Shape context and chamfer matching in cluttered scenes. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 127–133 (2003)
5. Ferrari, V., Tuytelaars, T., Gool, L.V.: Object detection by contour segment networks. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 14–28. Springer, Heidelberg (2006)
6. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of adjacent contour segments for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 36–51 (2008)

7. Ferrari, V., Jurie, F., Schmid, C.: Accurate object detection with deformable shape models learnt from images. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
8. Opelt, A., Pinz, A., Zisserman., A.: A boundary-fragment-model for object detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 575–588. Springer, Heidelberg (2006)
9. Leibe, B., Schiele, B.: Scale invariant object categorization using a scale-adaptive mean-shift search. In: Rasmussen, C.E., Bülthoff, H.H., Schölkopf, B., Giese, M.A. (eds.) DAGM 2004. LNCS, vol. 3175, pp. 145–153. Springer, Heidelberg (2004)
10. Shotton, J., Blake, A., Cipolla, R.: Multi-scale categorical object recognition using contour fragment. IEEE Transactions on Pattern Analysis and Machine Intelligence (2008)
11. Shotton, J., Blake, A., Cipolla, R.: Contour-based learning for object detection. In: IEEE International Conference on Computer Vision, pp. 503–510 (2005)
12. Wu, B., Nevatia, R.: Simultaneous object detection and segmentation by boosting local shape feature based classifier. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
13. Wu, Y., Si, Z., Fleming, C., Zhu, S.: Deformable template as active basis. In: IEEE International Conference on Computer Vision, pp. 1–8 (2007)
14. Felzenszwalb, P., Schwartz, J.: Hierarchical matching of deformable shapes. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
15. Basri, R., Costa, L., Geiger, D., Jacobs, D.: Determining the similarity of deformable shapes. Vision Research 38, 2365–2385 (1998)

Brain Hallucination

François Rousseau

LSIIT, UMR CNRS/ULP, Strasbourg, France

Abstract. In this paper, we investigate brain hallucination, or generating a high resolution brain image from an input low-resolution image, with the help of another high resolution brain image. Contrary to interpolation techniques, the reconstruction process is based on a physical model of image acquisition. Our contribution is a new regularization approach that uses an example-based framework integrating non-local similarity constraints to handle in a better way repetitive structures and texture. The effectiveness of our approach is demonstrated by experiments on realistic Magnetic Resonance brain images generating automatically high-quality hallucinated brain images from low-resolution input.

1 Introduction

In medical imaging, a so-called low resolution 3D image is a stack of 2D thick slices. As a result, 3D data are generally not isotropic. This paper describes a method to reconstruct a 3D image that has a higher spatial resolution than the original image. In medical imaging, this is usually done by applying interpolation techniques [11], which can be divided into two groups: scene-based and object-based methods [10]. Scene-based approaches use only image intensities to determine the interpolated intensity (for instance: nearest neighbor, linear interpolation, spline-based interpolation). Such scene-based methods produce perceptually unsatisfactory results with blurred edges and textures. Many edge-preserving interpolation techniques have been reported to handle this problem. However, these techniques rely on accurate edge information that is not obtainable from coarse data. In order to guide the interpolation process, object-based methods make use of additional information extracted from images. An example of an object-based method is the registration-based approach where non-rigid registration is used to register adjacent slices, and then interpolation is carried out between corresponding positions in each slice [8], [14]. However, scene-based and object-based techniques do not take advantage of a model of the imaging process.

Another approach for image up-sampling is the model-based technique which relies on modeling the imaging processes and using regularization methods describing *a priori* constraints. This approach is related to super-resolution (SR) whose the purpose is to combine low resolution (LR) images to produce an image that has a higher spatial resolution than the original images [3]. In medical imaging, several SR methods have been proposed to combine LR images to reconstruct one HR image [15]. SR is a large research field encompassing many

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 497–508, 2008.

applications. The work we describe in this paper is related to single-frame SR [16], meaning that one LR image is used to generate a high resolution (HR) image. More specifically, we focus on studies involving Magnetic Resonance (MR) imaging for which an anatomical HR image and several other LR images are acquired to keep acquisition time at an acceptable level for the patient (see Figure 1). This is the case for many MR acquisitions performed in routine such as follow-up of multiple sclerosis disease, brain tumor evolution or diffusion tensor imaging. Typically, one isotropic HR T1-weighted image and several anisotropic LR images (such as T2-weighted, FLAIR or proton density images) are acquired. In this context, we propose a new method which uses information from a HR image to aid in the image magnification of a LR image.

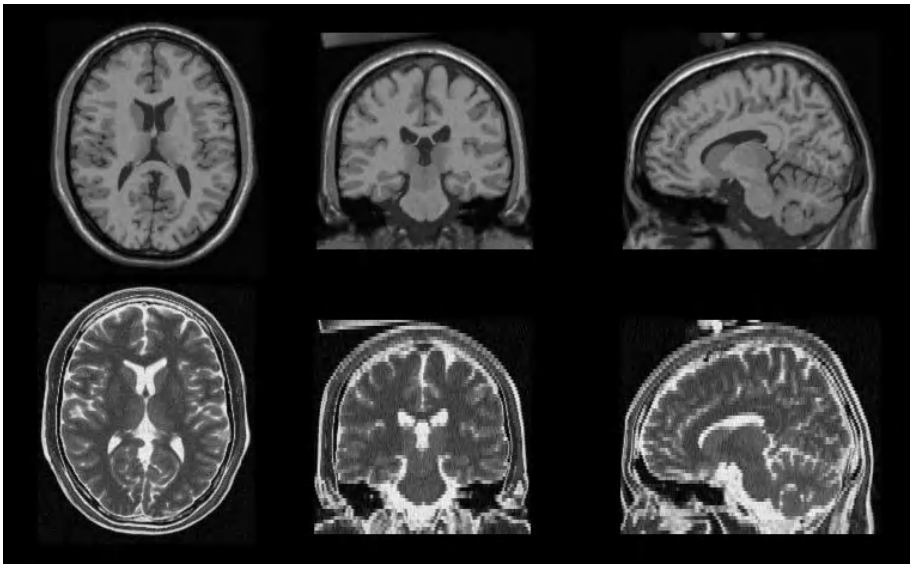


Fig. 1. Example of MR data [5]. First row: high resolution T1-weighted image ($1mm$ slice thickness); second row: low resolution T2-weighted image ($3mm$ slice thickness). The purpose of the method is to reconstruct a high-resolution T2-weighted image using information from the T1-weighted image.

2 Image Magnification

In this section, we present the model-based framework for image magnification which leads to an ill-posed inverse problem. Then, recently proposed regularization approaches relying on the example-based methodology are described.

2.1 Model-Based Framework

Contrary to interpolation approaches, we model the physical problem as in SR framework and the reconstructed image is obtained by inverting this problem. Model-based approaches use a generic observation model such as:

$$\mathbf{y} = DBW\mathbf{x} + \mathbf{n} \tag{1}$$

where $\mathbf{y}$ denotes the LR image, $\mathbf{x}$ is the high resolution (HR) image, $\mathbf{nr}$ represents observation noise, D is the subsampling matrix, B a blur matrix, W is the geometric transformation.

The three operators can be combined into a single matrix H: $H = DBW$. The matrix H thus incorporates motion compensation, degradation effects, and sub-sampling for the LR image $\mathbf{y}$. In this paper, we assume that the degradation operator H and the noise characteristics are known.

Based on this model, the SR image can be estimated by minimizing a least-square cost function such as:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{y} - H\mathbf{x}\|^2 \ . \tag{2}$$

For such inverse problem, some form of regularization plays a crucial role and must be included in the cost function to stabilize the problem or constrain the space of solutions. Thus, the HR image is computed by considering the following equation:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{y}, H) + \lambda\mathcal{R}(\mathbf{x}). \tag{3}$$

$\mathcal{L}(\mathbf{x}, \mathbf{y}, H)$ is a data fidelity term related to the physical model that penalizes inconsistency between the estimated HR image $\mathbf{x}$ and the observed LR image $\mathbf{y}$. $\mathcal{R}(\mathbf{x})$ is a regularization term. A common approach for regularization is to take explicitly into account the image geometry and to introduce a global weight λ that balances the contribution of prior smoothness terms and a fidelity term. Examples of pixel-based regularizers are Tikhonov regularization, Markov random field *a priori* image model or total variation.

However, there are two major drawbacks of pixel-based regularization methods: 1) there is no satisfying way to estimate the smoothing or regularization parameters from data, 2) generic smoothness priors may help regularize the problem, but cannot replace the missing information.

2.2 Example-Based Methods

To develop better image reconstruction algorithms, explicit models for the many regularities and geometries seen in local patterns are needed. In contrast to the pixel-based regularization approach, example-based methods consists in modeling non-local pairwise interactions from training data or a library of image patches. The principle of example-based SR methods is to add a similarity constraint between the voxels of HR image and the nearest examples present in the learning database [2],[7]. Such approach suggests that the reconstructed image should locally look like examples existing in the learning database. The regularization term can then be defined in a general way as follows:

$$\mathcal{R}(\mathbf{x}, \mathcal{E}) = \sum_{\mathbf{v},\mathbf{k}\in\Omega(\mathbf{v})} w_{\mathbf{v},\mathbf{k}} \|f(\mathbf{x}(\mathbf{v})) - \mathcal{E}_{\mathbf{v},\mathbf{k}}\|^2 \tag{4}$$

where $f(\mathbf{x}(\mathbf{v}))$ is an operator on the HR image $\mathbf{x}$ at the voxel $\mathbf{v}$, $\Omega(\mathbf{v})$ is a neighborhood of $\mathbf{v}$, $\mathcal{E}$ is the learning database, $\mathcal{E}_{\mathbf{v},\mathbf{k}}$ is a element of $\mathcal{E}$ related to $\mathbf{v}$ and $w_{\mathbf{v},\mathbf{k}}$ is a local weight. It is important to note that in example-based methods the regularization term $\mathcal{R}$ depends on the learning database $\mathcal{E}$.

Baker *et al.* in [2] have proposed a recognition-based gradient prior which enforces the constraints that the gradient of the HR image should be equal to the gradient of the best matching training image. The similarity constraint is then computed on image gradient values between each pixel of HR image and the best matching pixel in the learning database. In Equation 4, $f(\mathbf{x}(\mathbf{v}))$ would stand for the gradient value of the SR image $\mathbf{x}$ at the voxel $\mathbf{v}$ and $\mathcal{E}_{\mathbf{v},\mathbf{k}}$ would be the gradient value of the best matching pixel with respect to $\mathbf{v}$ in the learning database. Recently, Datsenko *et al.* in [7] have proposed another example-based regularization approach suggesting that the reconstructed image should agree with every found example and in every location. In this approach, $f(\mathbf{x}(\mathbf{v}))$ would be a patch of $\mathbf{x}$ centered in the voxel $\mathbf{v}$ and $\mathcal{E}_{\mathbf{v},\mathbf{k}}$ are the k nearest patches existing in the learning database.

In the context of face images, using a dedicated learning database, the problem of HR image reconstruction from a LR image has been investigated by Baker and Kanade [1]. This has been called *face hallucination* (see also [12]). Although the possibility to introduce new image priors makes the example-based approach very attractive, the key point (which can be a major drawback) is the need of a relevant learning database.

In this work, we proposed to take advantage of the MR imaging context and to investigate the use of a patch-based approach without any learning database by assuming that there exists related patterns in the LR anisotropic image and a HR isotropic image of the same patient.

3 Brain Hallucination

The purpose of this work is to reconstruct from one LR anistropic image with thick slices a HR isotropic brain image. In this context, we propose to use a patch-based approach to define the regularization term in order to take into account complex spatial interactions in images. Moreover, in contrast to example-based approaches for image modeling, the proposed method is unsupervised and thus uses no image patch learning database and no computational intensive training algorithms.

3.1 A New Regularization Term

The key idea of the proposed approach consists in saying that a HR image $\mathcal{E}_{\mathbf{x}}$ of a patient contains relevant examples which should be used to reconstruct a HR image $\mathbf{x}$ from a LR image $\mathbf{y}$ of the same patient and thus the HR image $\mathcal{E}_{\mathbf{x}}$ can be considered as a relevant candidate to be the learning database $\mathcal{E}$. We propose to introduce into the regularization term a similarity criterion between $\mathbf{x}$ and a patch-based regularized version of $\mathbf{x}$ using weights estimated from the HR image

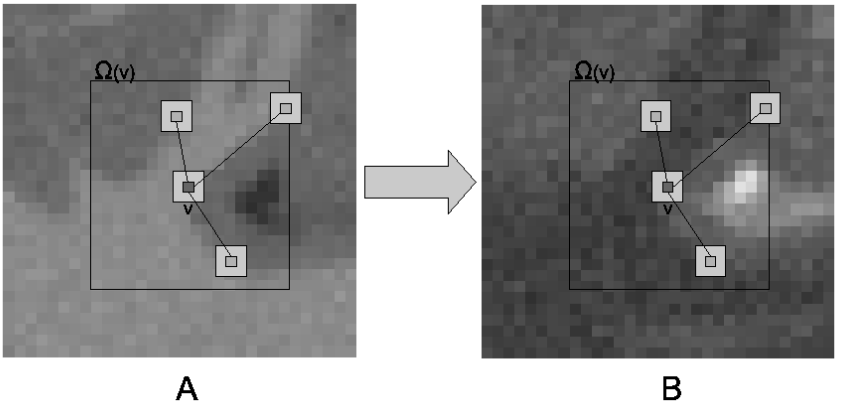


Fig. 2. Illustration of the regularization principle: the denoised value at voxel $\mathbf{v}$ in $\mathbf{x}$ is the weighted average of all intensities of voxels of $\mathbf{x}$ in the search volume $\Omega(\mathbf{v})$, based on the similarity of their intensity neighborhoods in the example HR image $\mathcal{E}_\mathbf{x}$. A) the example HR image $\mathcal{E}_\mathbf{x}$ is used to compute the weigths $w_{NLM}(\mathbf{v}, \mathbf{k}, \mathcal{E}_\mathbf{x})$. These weights describe local interactions between 3D patches. B) reconstructed HR image $\mathbf{x}$: weights $w_{NLM}(\mathbf{v}, \mathbf{k}, \mathcal{E}_\mathbf{x})$ estimated using the example HR image $\mathcal{E}_\mathbf{x}$ are used to estimated a regularized version of $\mathbf{x}$.

$\mathcal{E}_\mathbf{x}$. This regularization term can be seen as a similarity constraint between local patterns of $\mathbf{x}$ and $\mathcal{E}_\mathbf{x}$.

In the proposed approach, examples present in HR image $\mathcal{E}_\mathbf{x}$ are introduced in the regularization term by considering a denoised version of the HR image $\mathbf{x}$ obtained with a example-based framework. The proposed regularization term is defined as follows:

$$\mathcal{R}(\mathbf{x}, \mathcal{E}) = \sum_{\mathbf{v}} w_\mathcal{R}(\mathbf{v}) \|\mathbf{x}(\mathbf{v}) - d_{NLM}(\mathbf{x}(\mathbf{v}), \mathcal{E}_\mathbf{x})\|^2 \tag{5}$$

where $w_\mathcal{R}(\mathbf{v})$ is a local weight between data term $\mathcal{L}$ and regularization term $\mathcal{R}$. $d_{NLM}(\mathbf{x}(\mathbf{v}), \mathcal{E}_\mathbf{x})$ is a denoised (or regularized) version of $\mathbf{x}(\mathbf{v})$, estimated as follows:

$$d_{NLM}(\mathbf{x}(\mathbf{v}), \mathcal{E}_\mathbf{x}) = \sum_{\mathbf{k} \in \Omega(\mathbf{v})} w_{NLM}(\mathbf{v}, \mathbf{k}, \mathcal{E}_\mathbf{x}) \mathbf{x}(\mathbf{k}) \tag{6}$$

where $\Omega(\mathbf{v})$ corresponds to the neighborhood of the voxel $\mathbf{v}$ in the HR image $\mathbf{x}$. Figure 2 shows how example patterns in HR image $\mathcal{E}_\mathbf{x}$ are used to compute a regularized version of the reconstructed HR image $\mathbf{x}$. The use of a non-local approach to define the regularization term has the advantage over the PDE approach to handle in a better way repetitive structures and texture. As far as we know, our approach is the first one using a non-local operator for regularization to constraint the reconstruction process.

Our work is related to the Non Local Means (NLM) method introduced by Buades *et al.* in [4] for image denoising and recently applied for MRI denoising by Coupé *et al.* [6]. Buades *et al.* have shown that, for 2D natural images, the NLM

filter outperforms state-of-the-art denoising methods such as the Rudin-Osher-Fatemi Total Variation minimization scheme or the Perona-Malik Anisotropic diffusion. In the NLM algorithm, the restored intensity of the voxel $\mathbf{v}$, $NLM(\mathbf{v})$, is a weighted average of all voxel intensities in the image I:

$$NLM(\mathbf{v}) = \sum_{\mathbf{k} \in I} w_{NLM}(\mathbf{v}, \mathbf{k}) I(k) \tag{7}$$

where $I(\mathbf{k})$ is the intensity at voxel $\mathbf{k}$ and $w_{NLM}(\mathbf{v}, \mathbf{k})$ is the weight assigned to $I(\mathbf{k})$ in the restoration at voxel $\mathbf{v}$. The NLM method tries to take advantage of the high degree of redundancy of any natural image and appears to be an unsupervised example-based denoising method.

3.2 How to Deal with Outliers ?

The regularization term proposed in Equation 5 has been obtained by assuming that local patterns in the HR image $\mathcal{E}_{\mathbf{x}}$ could be used as examples to regularize the reconstructed HR image $\mathbf{x}$. However, there are some cases where this assumption may not hold. In the context of MR imaging, images $\mathcal{E}_{\mathbf{x}}$ and $\mathbf{y}$ are not acquired with the same MR sequence. Multimodal MR data do not reveal the same tissue specificity; for instance, Multiple Sclerosis (MS) lesions can be clearly visible in T2-weighted images but not in T1-weighted images (see Figure 3). In this case, the HR T1-weighted image may not be a relevant candidate to guide the reconstruction process. In order to handle the case of possible outliers, we propose to modify the regularization term by locally analyzing the correlation between two sets of local patterns.

To handle outliers, we propose another adaptive regularization term by modifying the way to compute the denoised version of the HR reconstructed image $\mathbf{x}$:

$$d_{NLM}(\mathbf{x}(\mathbf{v}), \mathcal{E}_{\mathbf{x}}) = \alpha \sum_{\mathbf{k} \in \Omega(\mathbf{v})} w_{NLM}(\mathbf{v}, \mathbf{k}, \mathcal{E}_{\mathbf{x}}) \mathbf{x}(\mathbf{k}) + (1-\alpha) \sum_{\mathbf{k} \in \Omega(\mathbf{v})} w_{NLM}(\mathbf{v}, \mathbf{k}, \mathbf{x}) \mathbf{x}(\mathbf{k}) \tag{8}$$

$d_{NLM}(\mathbf{x}(\mathbf{v}), \mathcal{E}_{\mathbf{x}})$ is now a weighted average of two denoised version of $\mathbf{x}(\mathbf{v})$. The first term is the denoised version of $\mathbf{x}(\mathbf{v})$ computed by using the HR image $\mathcal{E}_{\mathbf{x}}$ and the second term is also a denoised version of $\mathbf{x}(\mathbf{v})$ obtained with the current estimate of the reconstructed HR image $\mathbf{x}$. The weight α is defined as the correlation between the two set of weights $w_{NLM}(\mathbf{v}, \mathbf{k}, \mathcal{E}_{\mathbf{x}})$ and $w_{NLM}(\mathbf{v}, \mathbf{k}, \mathbf{x})$. If the weights are correlated, the HR image $\mathcal{E}_{\mathbf{x}}$ is likely to be a relevant candidate to guide the reconstruction process and thus, α is close to 1. If the weights are uncorrelated, the presence of outliers is detected and α is then close to 0. Defining $d_{NLM}(\mathbf{x}(\mathbf{v}), \mathcal{E}_{\mathbf{x}})$ in such way allows to choose the best examples to regularize the reconstructed HR image $\mathbf{x}$.

3.3 Computation of Weights w_{NLM}

Weights $w_{NLM}(\mathbf{v}, \mathbf{k}, \mathcal{E}_{\mathbf{x}})$ and $w_{NLM}(\mathbf{v}, \mathbf{k}, \mathbf{x})$ are computed in the same way. For the sake of clarity, we describe only the computation of $w_{NLM}(\mathbf{v}, \mathbf{k}, \mathcal{E}_{\mathbf{x}})$.

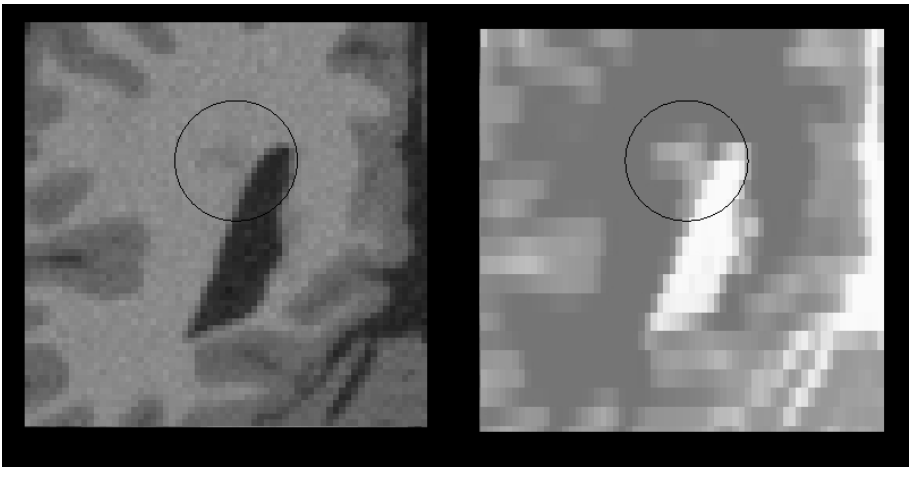


Fig. 3. Illustration of possible outliers. The MS lesion (in the circle) is less visible in the HR T1-weighted image than in the LR T2-weighted image. In this case, the assumption that local patterns in the HR image $\mathcal{E}_{\mathbf{x}}$ could be used as examples to regularize the reconstructed HR image $\mathbf{x}$ does not hold.

As in Coupé *et al.* [6], to reduce time computation, weight $w_{NLM}(\mathbf{v}, k, \mathcal{E}_{\mathbf{x}})$ is computed as follows:

$$w_{NLM}(\mathbf{v}, \mathbf{k}, \mathcal{E}_{\mathbf{x}}) = \frac{1}{Z_{\mathbf{v}}} e^{-\frac{\|P_{3D}(\mathcal{E}_{\mathbf{x}}(\mathbf{v})) - P_{3D}(\mathcal{E}_{\mathbf{x}}(\mathbf{k}))\|_2^2}{2\beta\hat{\sigma}^2|N_i|}} \tag{9}$$

where $P_{3D}(\mathcal{E}_{\mathbf{x}}(\mathbf{k}))$ is a 3D patch of HR image $\mathcal{E}_{\mathbf{x}}$ centered in voxel k and $P_{3D}(\mathcal{E}_{\mathbf{x}}(\mathbf{v}))$ is the 3D patch of HR image $\mathcal{E}_{\mathbf{x}}$ centered in voxel $\mathbf{v}$; $Z_{\mathbf{v}}$ is a constant of normalization. The distance between the 3D patches is the sum over voxels of patches of intensity differences using $L2$ norm. With the assumption on Gaussian noise (Equation 1), β is set to 1 (see [4] for theoretical justifications) and the standard deviation of noise is estimated via pseudo-residuals ϵ_v as defined in [9]. For each voxel $\mathbf{v}$ of HR image $\mathcal{E}_{\mathbf{x}}$, let us define:

$$\epsilon_v = \sqrt{\frac{6}{7}} \left(\mathcal{E}_{\mathbf{x}}(\mathbf{v}) - \frac{1}{6} \sum_{\mathbf{k} \in \Omega(\mathbf{v})} \mathcal{E}_{\mathbf{x}}(\mathbf{v}) \right) \tag{10}$$

where $\Omega(\mathbf{v})$ is the 6-neighborhood at voxel $\mathbf{v}$. The standard deviation of noise is computed as the least square estimator:

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{k=1}^{n} \epsilon_k^2 \tag{11}$$

where n is the number of voxel in the HR image.

Moreover, as suggested by Mahmoudi and Sapiro in [13], voxel preselection can avoid useless computation and also improve the result of denoising. As in

[6], the preselection of relevant voxel is based on the mean and variance of 2D patches. $w_{\mathbf{v},k}$ is set to 0 if one of these conditions is not fulfilled:

$$\mu < \frac{\overline{P_{2D}(\mathbf{x}(\mathbf{v}))}}{\overline{P_{2D}(\mathbf{y}(\mathbf{k}))}} < \frac{1}{\mu} \tag{12}$$

$$\sigma^2 < \frac{var(P_{2D}(\mathbf{x}(\mathbf{v})))}{var(P_{2D}(\mathbf{y}(\mathbf{k})))} < \frac{1}{\sigma^2} \tag{13}$$

with $\mu = 0.95$ and $\sigma^2 = 0.5$.

3.4 Balance between Fidelity Data Term and Regularization Term

In Equation 3, λ is a global weight between fidelity data term and regularization term. This weight is usually tuned by error and trial since there is no satisfying way to estimate it. In our approach, λ is set to 1 and we propose a non-stationary approach by using local weights $w_{\mathcal{R}}(\mathbf{v})$ for each voxel $\mathbf{v}$ which are defined using the point spread function of the acquisition system as follows:

$$w_{\mathcal{R}}(\mathbf{v}) = \frac{1 - \sum_r b(\|\mathbf{v} - \mathbf{y}_r\|_2)}{\sum_r b(\|\mathbf{v} - \mathbf{y}_r\|_2)} \tag{14}$$

where b is the point spread function (b is related to the blur matrix B used in Equation 1). $w_{\mathcal{R}}(\mathbf{v})$ increases if the voxel $\mathbf{v}$ is far of LR image data.

4 Results

In each experiment, as suggested in [6], parameter values for patches are: search area $11 \times 11 \times 11$ voxels, patch size $3 \times 3 \times 3$ voxels. Moreover, a gradient descent method is used to optimize the cost function.

To explore the ability to reconstruct high resolution image of realistic typical anatomical brain structures, we applied the algorithm on MRI images of Brainweb [5]. Brainweb is a simulated brain database which is often used as a gold standard for the analysis of in vivo acquired data. The database contains simulated brain MRI data based on two anatomical models: normal and multiple sclerosis (MS lesions have been extracted from real MRI data). Using the HR image provided by Brainweb, we have generated low resolution images using the observation model described by Equation 1. Thus, the ground truth is available and it can be compared with the reconstructed HR brain images. Figures 4 and 5 show the results for pathological and non pathological MR Brainweb images for axial, coronal and sagittal views. We also reported PSNR in decibels (dB) results obtained with the different methods in Table 1.

$$PSNR = 10 \log_{10} \left(\frac{d^2}{|\Omega|^{-1} \sum_{\mathbf{v} \in \Omega} (\mathbf{x}(\mathbf{v}) - \hat{\mathbf{x}}(\mathbf{v}))^2} \right)$$

where d is the reference image dynamic. Results obtained with the proposed approach compare favorably with fifth-order B-spline interpolation. Fine details have been successfully recovered and contrast between structures has been improved.

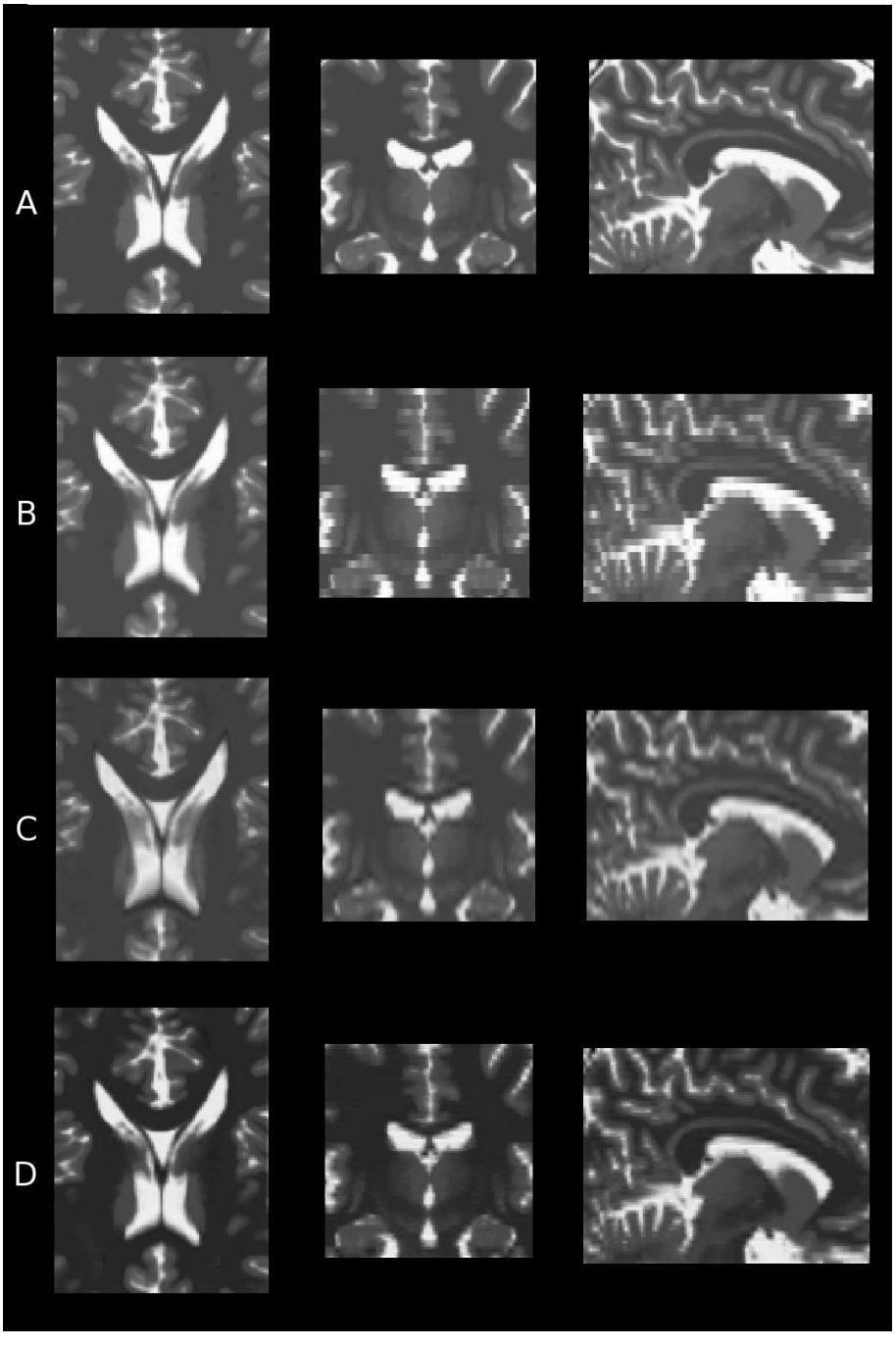


Fig. 4. Reconstruction results (non pathological case). A) Ground truth, B) input LR image, C) fifth order B-spline interpolation, D) the proposed approach.

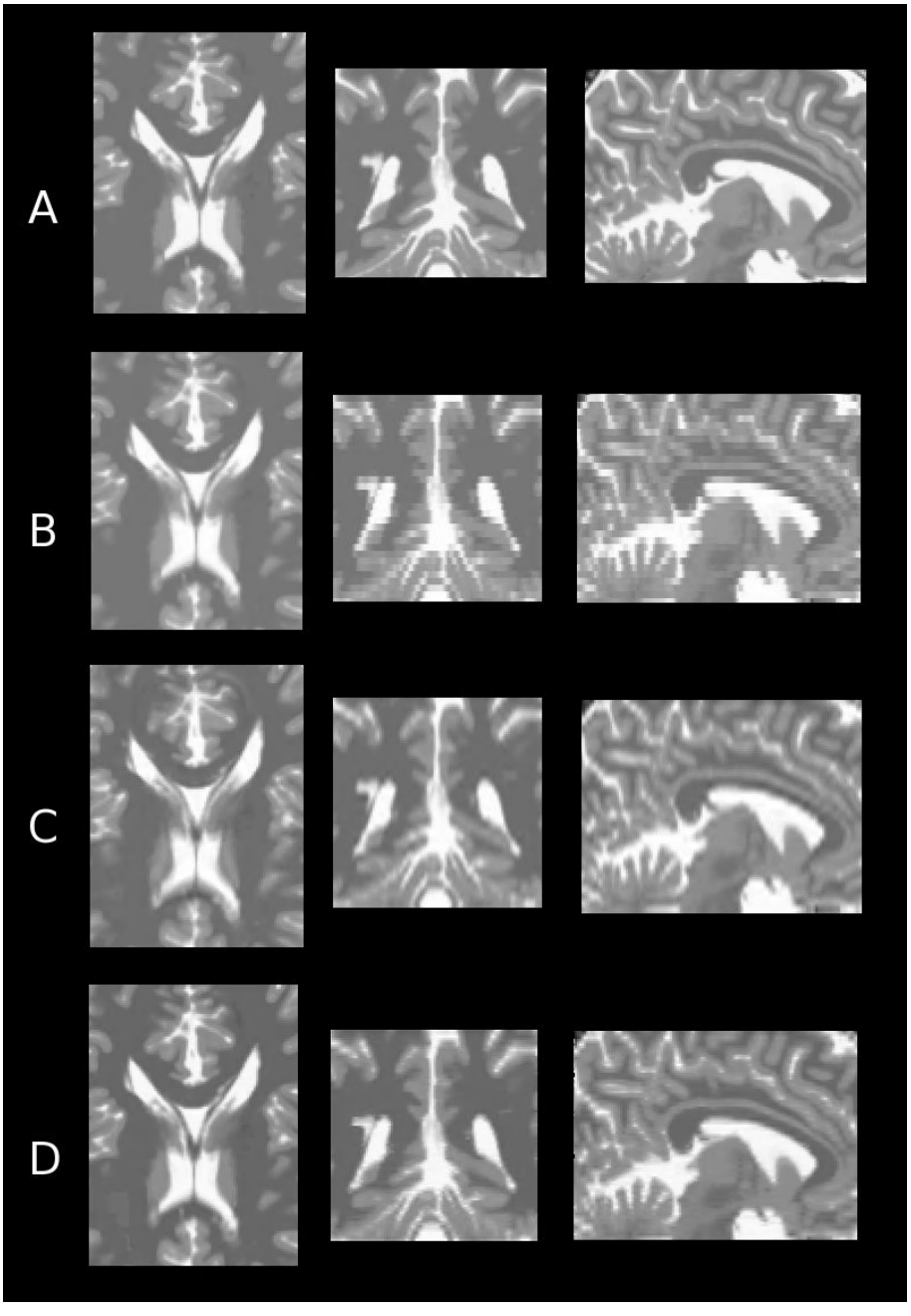


Fig. 5. Reconstruction results in presence of MS lesions. A) Ground truth, B) input LR image, C) fifth order B-spline interpolation, D) the proposed approach.

Table 1. Performances of image reconstruction methods

Method	Non pathological Image	Multiple Sclerosis Image
Nearest Neighbour Interpolation	23.26	26.11
Trilinear Interpolation	26.17	27.30
3rd order B-spline Interpolation	27.80	28.36
5th order B-spline Interpolation	27.93	28.48
Our method	28.24	29.75

5 Conclusion

We have shown that example-based approaches such as non-local means can be embedded into the reconstruction process to enhance the performance of super resolution techniques. If a HR brain image of the patient is available, LR images of the same patient can be enhanced by exploiting non-local pairwise interactions. Experimental results prove that the developed algorithm including an image acquisition model compares favorably with interpolation approach. High resolution imaging is a key point for MR brain image analysis in order to study anatomical details. Such HR image reconstruction algorithm represents an important step towards multimodal brain analysis at fine scale.

To develop better image reconstruction algorithms, example-based SR methods try to add a similarity constraint between the voxels of HR image and the nearest examples present in the learning database. In the medical context studied in this paper, the learning database contains only one HR image. Further work concerns the extension and application of this approach to other domains such as computational photography or image fusion, and also when no HR image example is available.

Acknowledgment

The research leading to these results has received funding from the European Research Council under the European Communitys Seventh Framework Programme (FP7/2007-2013 Grant Agreement no. 207667). The author would like to thank S. Faisan from LSIIT/University of Strasbourg for fruitful discussions.

References

1. Baker, S., Kanade, T.: Hallucinating Faces. In: Fourth Int. Conf. on Automatic Face and Gesture Recognition (2000)
2. Baker, S., Kanade, T.: Limits on Super-Resolution and How to Break Them. IEEE Trans. Pattern Analysis and Machine Intelligence 24(9), 1167–1183 (2002)
3. Bose, N.K., Chan, R.H., Ng, M.K.: Special Issue: High Resolution Image Reconstruction. Int. J. of Imaging Systems and Technology 14(2-3) (2004)
4. Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms, with a new one. Multiscale Modeling & Simulation 4(2), 490–530 (2005)

5. Cocosco, C.A., Kollokian, V., Kwan, R.K.-S., Evans, A.C.: BrainWeb: Online Interface to a 3D MRI Simulated Brain Database. In: Proceedings of 3-rd International Conference on Functional Mapping of the Human Brain, vol. 5(4) (1997)
6. Coupé, P., Yger, P., Prima, S., Hellier, P., Kervrann, C., Barillot, C.: An Optimized Blockwise Non Local Means Denoising Filter for 3D Magnetic Resonance Images. IEEE Trans. Medical Imaging (2007)
7. Datsenko, D., Elad, M.: Example-based single document image super-resolution: a global MAP approach with outlier rejection. Multidim. Syst. Sign. Process 18, 103–121 (2007)
8. Frakes, D.H., Dasi, L.P., Pekkan, K., Kitajima, H.D., Sundareswaran, K., Yoganathan, A.P., Smith, M.J.T.: A New Method for Registration-Based Medical Image Interpolation. IEEE Trans. Medical Imaging 27(3), 370–377 (2008)
9. Gasser, T., Sroka, L., Steinmetz, C.: Residual variance and residual pattern in nonlinear regression. Biometrika 73(3), 625–633 (1986)
10. Grevera, G.J., Udupa, J.K.: An objective comparison of 3-D image interpolation methods. IEEE Transactions on Medical Imaging 17, 642–652 (1998)
11. Lehmann, T., Gonner, C., Spitzer, K.: Survey: Interpolation Methods in Medical Image Processing. IEEE Transactions on Medical Imaging 18(11), 1049–1075 (1999)
12. Liu, C., Shum, H.-Y., Freeman, W.T.: Face Hallucination: Theory and Practice. Int. Journal of Computer Vision 75(1), 115–134 (2007)
13. Mahmoudi, M., Sapiro, G.: Fast image and video denoising via nonlocal means of similar neighborhoods. IEEE Signal Processing Letters 12(12), 839–842 (2005)
14. Penney, G.P., Schnabel, J.A., Rueckert, D., Viergever, M.A., Niessen, W.J.: Registration-Based Interpolation. IEEE Transactions on Medical Imaging 23(7), 922–926 (2004)
15. Rousseau, F., Glenn, O., Iordanova, B., Rodriguez-Carranza, C., Vigneron, D., Barkovich, J., Studholme, C.: Registration-Based Approach for Reconstruction of High-Resolution in Utero Fetal MR Brain images. Academic Radiology 13(9), 1072–1081 (2006)
16. van Ouwerkerk, J.D.: Image super-resolution survey. Image and Vision Computing 24, 1039–1052 (2006)

Range Flow for Varying Illumination

Tobias Schuchert[1], Til Aach[2], and Hanno Scharr[1]

[1] Institute for Chemistry and Dynamics of the Geosphere, ICG-3: Phytosphere, Forschungszentrum Jülich, Germany
{t.schuchert,h.scharr}@fz-juelich.de
[2] Institute of Imaging & Computer Vision, RWTH Aachen University, Germany
til.aach@lfb.rwth-aachen.de

Abstract. In this paper range flow estimation is extended to handle brightness changes in image data caused by inhomogeneous illumination. Standard range flow computes 3d velocity fields from range and intensity image sequences. To this end it combines a depth change model and a brightness constancy model. In this contribution, the brightness constancy model is exchanged by (1) a gradient constancy model, (2) a combination of gradient and brightness constancy constraint that has been used successfully for optical flow estimation in literature, and (3) a physics-based brightness change model. Insensitivity to brightness changes can also be achieved by prefiltering of the input intensity data. High pass or homomorphic filtering are the most well known approaches from literature. In performance tests therefore the well known version and the novel versions of range flow estimation are investigated on prefiltered or non-prefiltered data using synthetic ground-truth and real data from a botanical experiment.

1 Introduction

In this paper influences of brightness models in 3d velocity field estimation are investigated. The brightness model is only one module of usual motion estimation methods. Typical methods consist of data prefiltering, brightness constraint equations describing imaging physics locally (see e.g. [1,2]) calculated by suitable convolution filtering [3,4], parameter estimation scheme like local least squares [5], local total least squares [6], or variational approaches [7]. Occlusions and other model violations are typically handled using robust error norms instead of plain least squares [8,9]. Variational estimators in addition allow for incorporation of prior knowledge, e.g. in form of regularization terms closing holes or reducing so-called aperture problems.

Motivation of this work is a target application, namely plant growth estimation. Growth is one of the most important processes in plant life and therefore of high botanical interest. However, this paper does not focus on a best estimation *system* for this application, but isolates influences of brightness change models frequently occurring in such botanical but also other data.

A lot of work has been carried out on estimating 3d motion fields. Range flow estimation [10,11] uses data solely from range sensors whereas [12,13] incorporate information from both range and image sensors. Reconstruction of

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 509–522, 2008.

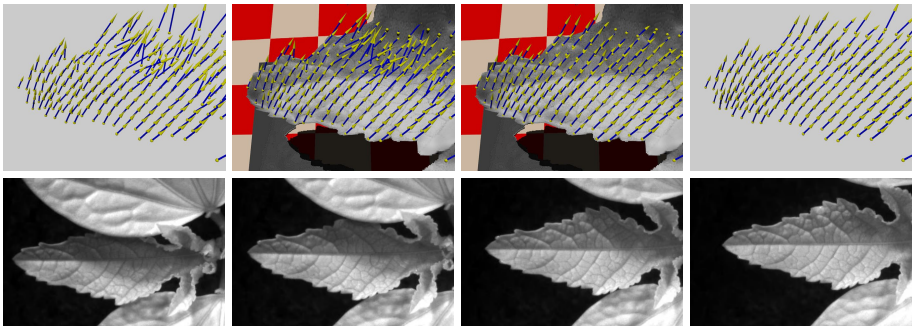


Fig. 1. Castor bean plant leaf. Top: Estimated motion vector fields. The two images to the left and to the right show the same results, respectively, but with and without 3d structure shown. Left images: standard range flow. Right: proposed TAYLOR model. Bottom: Images of the input sequence.

scene flow and 3d structure from the observed optical flow in several cameras has been proposed by [14,15,16]. These scene flow and most optical flow based approaches [17,18] imply brightness constancy and are therefore not suitable if substantial brightness changes are present in a sequence (cmp. Fig. 1, upper part of the shown leaf). For optical flow estimation less brightness sensitive models, e.g. constancy of intensity gradient vector [19,20] have been proposed, as well as physics-based brightness change models [21,1]. The physics-based models of [1] have been recently adapted and extended for moving surfaces under inhomogeneous illumination [2]. Suppressing brightness changes by prefiltering the intensity data has been shown to be very efficient. One of the simplest approaches is applying a spatial high-pass filter to minimize the effect of global brightness inhomogeneities. Toth et al [22] show that using homomorphic prefilters [23] can highly improve motion detection in image sequences with inhomogeneous illumination. More recent approaches for scene flow estimation use statistical similarity measures [24], a gradient constancy constraint [25] or probability distributions for optical flow and disparity [26] to make scene flow estimation more robust against brightness changes.

Our contribution. The contribution of this paper is twofold: On the one hand range flow estimation is extended to be able to handle inhomogeneous illumination. To this end the techniques known for optical flow estimation are introduced in the range flow constraints, namely (1) gradient constancy [19] and (2) mixture of brightness and gradient constancy [20] as well as physics-based brightness modeling [2]. On the other hand performance of standard and the novel range flow models is investigated. To this end the models are tested on two kinds of synthetic data sets with ground truth available. These sequences are either used with or without suppression of illumination inhomogeneities by high-pass or homomorphic filtering. A motion estimation result for a "real" image sequence from a botanical experiment on leaf growth is shown in Fig. 1. Performance experiments focus on and therefore are especially designed for *brightness model influences*.

Input data without occlusions, inner borders, holes or aperture problems has been selected. Thus *no robust estimators* handling occlusion or *variational estimators* closing holes are needed, but local least squares estimation on a large neighborhood is suitable. In a final system robust and/or variational estimators may be used of course if needed, but not for investigations of *model* influences.

Paper organization. In Sec. 2 we derive the differential range flow model. Then different prefilters and intensity constraints in the context of range flow are presented in Sec. 3. In Sec. 4 we briefly review parameter estimation followed by experiments on synthetic and real data in Sec. 5.

2 Range Flow

Range flow as used here is based on two motion constraints: one for range data and one for intensity data. Following [13], we briefly derive these two constraints in this section.

2.1 Range Constraint

Let a surface be described by its depth Z as a function of space and time $Z = Z(X, Y, t)$, where X, Y and Z are spatial coordinates and t denotes time. We select X and Y being aligned with camera sensor coordinates x and y, respectively. Z-axis points along the principal axis of the assumed projective camera. The total derivative of Z with respect to time then yields the so called range flow motion constraint equation

$$\frac{dZ}{dt} = \partial_X Z \frac{dX}{dt} + \partial_Y Z \frac{dY}{dt} + \partial_t Z \tag{1}$$

where partial derivatives are denoted as e.g. $\partial_X Z = \frac{\partial Z}{\partial X}$. Range flow is defined to be $f = [U, V, W]^{\mathrm{T}} := [\frac{dX}{dt}, \frac{dY}{dt}, \frac{dZ}{dt}]^{\mathrm{T}}$.

Range data is given as data sets $X = X(x, y, t)$, $Y = Y(x, y, t)$ and $Z = Z(x, y, t)$ on the sampling grid. Partial derivatives are not computed on world coordinate data but directly on the sensor grid in order to avoid interpolation artifacts and expensive preprocessing steps. Range flow, i.e. the total derivatives of the world coordinates with respect to time may then be calculated as

$$U = \partial_x X \dot{x} + \partial_y X \dot{y} + \partial_t X \tag{2}$$

$$V = \partial_x Y \dot{x} + \partial_y Y \dot{y} + \partial_t Y \tag{3}$$

$$W = \partial_x Z \dot{x} + \partial_y Z \dot{y} + \partial_t Z \tag{4}$$

where total derivatives with respect to time are indicated by a dot. Being not interested in the change on the sensor grid, i.e. optical flow, $\dot{x}$ and $\dot{y}$ can be eliminated. This yields

$$\frac{\partial(Z, Y)}{\partial(x, y)} U + \frac{\partial(X, Z)}{\partial(x, y)} V + \frac{\partial(Y, X)}{\partial(x, y)} W + \frac{\partial(X, Y, Z)}{\partial(x, y, t)} = 0 \tag{5}$$

where e.g.

$$\frac{\partial(Z,Y)}{\partial(x,y)} = \begin{vmatrix} \partial_x Z & \partial_x Y \\ \partial_y Z & \partial_y Y \end{vmatrix} = \partial_x Z \partial_y Y - \partial_y Z \partial_x Y \tag{6}$$

is the Jacobian of Z, Y with respect to x, y. Equation (5) only depends on derivatives in sensor coordinates and can be calculated easily using derivative kernels. Assuming aligned world and sensor coordinate systems ($\partial_y X = \partial_x Y = 0$) (5) reduces to

$$\begin{aligned}(\partial_y Y \partial_x Z)U + (\partial_x X \partial_y Z)V - (\partial_x X \partial_y Y)W \\ + (\partial_x X \partial_y Y \partial_t Z - \partial_x X \partial_t Y \partial_y Z - \partial_t X \partial_y Y \partial_x Z) = 0\,. \end{aligned} \tag{7}$$

2.2 Intensity Constraint

The range flow constraint is solely for range data and full flow can only be estimated for corners or point-like structures. Plant surfaces are often nearly planar, smooth surfaces resulting in aperture problems almost everywhere when using solely the range flow constraint. As proposed in [12] intensity data should be incorporated. Let the intensity of a point remain constant over the observation time interval. Then the so-called brightness constancy constraint equation often used for optical flow estimation (cmp. e.g. [17]) is valid. Linearization of this constraint yields

$$\frac{dI}{dt} = \partial_x I \dot{x} + \partial_y I \dot{y} + \partial_t I = 0\,. \tag{8}$$

Eliminating optical flow $(\dot{x}, \dot{y})$ using (2) and (3) yields

$$\frac{\partial(I,Y)}{\partial(x,y)}U + \frac{\partial(X,I)}{\partial(x,y)}V + \frac{\partial(X,Y,I)}{\partial(x,y,t)} = 0\,. \tag{9}$$

The estimated motion $[U, V, W]$ has to fulfill both constraints, the range flow constraint (5) and the intensity constraint (9). The intensity constraint more reliably yields point-to-point correspondences and therefore often solves the aperture problem. Together with the range constraint it allows also to solve for the vertical motion W. Combining range and intensity constraint and estimation of f via total least squares is shown in Sec. 4.

3 Handling Brightness Changes

Range flow estimation as presented in the previous section yields good results for objects under homogeneous, diffuse illumination. Problems occur for directed, inhomogeneous illumination, because the intensity constraint (9) is not well fulfilled anymore. In the following section different approaches to handle illumination changes are presented. Three of them lead to constraints novel in range flow estimation.

3.1 Prefiltering

A well known technique for illumination change suppression is suitable prefiltering making data illumination invariant.

Temporal and/or spatial high-pass filtering eliminates slow brightness changes in the data. However faster illumination changes both in spatial and temporal domain still remain in the data. In our experiments we test high-pass filtering with the filter

$$\tilde{I} = (1 - B_{11}) * I \tag{10}$$

where B_{11} denotes a spatial 11-tab binomial filter (see e.g. [27]) and $*$ is convolution.

A more sophisticated approach is using a homomorphic filter [23]. Following [22] we briefly derive a simple implementation of homomorphic filtering, which proved to be very successful in suppressing illumination changes. Homomorphic filtering uses the fact that image intensity $I(x, y)$ is proportional to incident illumination intensity $E(x, y)$, which is reflected by object surfaces with reflectance $R(x, y)$ in the observed scene. For Lambertian surfaces image intensity can be modeled as

$$I(x, y) \propto E(x, y) \cdot R(x, y) \; . \tag{11}$$

Reflectance R is the desired component for motion estimation as this part contains the structure of the scene and is temporally invariant. The logarithm transforms the multiplicative relation between illumination intensity E and reflectance R into an additive one

$$log(I(x, y)) \propto log(E(x, y)) + log(R(x, y)) \; . \tag{12}$$

Ideally $log(E)$ and $log(R)$ are separated in frequency domain as E is assumed to be low-frequent and R mainly high-frequent. In practice the two components overlap. Thus a tradeoff between suppressing brightness changes and loss of signal has to be made. However high-pass filtering more efficiently suppresses the illumination component after taking the logarithm of the signal. However, as the nonlinear log-operation makes the camera noise variance signal-dependent, therefore influencing parameter estimation, exponentiation returns an approximation of the sought for reflectance component.

3.2 Gradient Constancy Constraint

The intensity constraint (9) can be exchanged by an illumination invariant or insensitive constraint instead of or in addition to prefiltering the data.

A method reported to be successful in the pure optical flow case [20] is to assume that 2d image intensity gradient should not change along the motion trajectory. Derivative filtering is a highpass operation therefore also suppressing illumination changes (see Sec. 3.1). This leads to two linearized gradient constancy constraints

$$\frac{dI_x}{dt} = \partial_x I_x \dot{x} + \partial_y I_x \dot{y} + \partial_t I_x = 0 \tag{13}$$

$$\frac{dI_y}{dt} = \partial_x I_y \dot{x} + \partial_y I_y \dot{y} + \partial_t I_y = 0 \tag{14}$$

where lower indices indicate partial derivatives, e.g. $I_x := \partial_x I$, as before. Analogous to derivation of (9) optical flow $(\dot{x}, \dot{y})$ is eliminated using (2) and (3)

$$\frac{\partial(I_x, Y)}{\partial(x, y)} U + \frac{\partial(X, I_x)}{\partial(x, y)} V + \frac{\partial(X, Y, I_x)}{\partial(x, y, t)} = 0 \tag{15}$$

$$\frac{\partial(I_y, Y)}{\partial(x, y)} U + \frac{\partial(X, I_y)}{\partial(x, y)} V + \frac{\partial(X, Y, I_y)}{\partial(x, y, t)} = 0 \,. \tag{16}$$

3.3 Combined Intensity and Gradient Constancy Constraint

A known drawback of the gradient constancy constraint proposed in Sec. 3.2 is that it noticeably reduces the structure in the image and leads to aperture problems. Using both the intensity constraint and the gradient constraint simultaneously has reduced this effect in the optical flow case [20]. This leads to three constraint equations ((9), (15) and (16)) which should be fulfilled simultaneously for the horizontal and vertical range flow components U and V.

3.4 Physics Based Brightness Change Model

A different approach to handle brightness changes is to model them explicitly and estimate both optical flow and brightness change parameters. Haussecker and Fleet [1] proposed a generalized formulation of optical flow estimation based on models of brightness variation that are caused by time-dependent physical processes. Brightness changes along a temporal trajectory $\mathbf{x}(t) = (x(t), y(t))^{\mathrm{T}}$. This is described by a parameterized function h_I

$$I(\mathbf{x}(t), t) = h_I(I_0, t, \mathbf{a}) \tag{17}$$

where $I_0 = I(\mathbf{x}(0), 0)$ denotes image intensity at time $t = 0$ and $\mathbf{a} = [a_1, ...a_n]^{\mathrm{T}}$ contains n brightness change parameters. Taking the total derivative on both sides yields

$$\partial_x I \dot{x} + \partial_y I \dot{y} + \partial_t I = \dot{h}_I(I_0, t, \mathbf{a}) \,. \tag{18}$$

Assuming brightness constancy, i.e. $h_I(I_0, t, \mathbf{a}) = c$, (18) reduces to (8). Given a physical model h for brightness changes both the optical flow $(\dot{x}, \dot{y})$ and the parameter vector $\mathbf{a}$ need to be estimated. Several time-dependent illumination change models are proposed in [1], i.e. changing surface orientation, motion of the illuminant, and physical models of heat transport in infrared images. We use a brightness change model presented in [2] which handles spatially varying time-dependent illumination changes coming from directed, inhomogeneous illumination and changing surface orientation. In [2] the brightness change function is set to

$$h_I(I_0, t, \mathbf{a}) = I_0 \exp\{h(\Delta X, \Delta Y, t, \mathbf{a})\} \,. \tag{19}$$

The incident irradiance caused by the moving illuminant is assumed to be spatially inhomogeneous, therefore changing not only by a time dependent factor, but also varying smoothly in space. Approximating these brightness changes by a second order Taylor series yields

$$h(\Delta X, \Delta Y, t, \mathbf{a}) := \sum_{i=1}^{2} \left(a_i + a_{i,x}\Delta X + a_{i,y}\Delta Y\right) t^i \tag{20}$$

with spatial neighborhood $\Delta X, \Delta Y$ and temporal derivative

$$\dot{h}(\Delta X, \Delta Y, t, \mathbf{a}) = \sum_{i=1}^{2} i \left(a_i + a_{i,x}\Delta X + a_{i,y}\Delta Y\right) t^{i-1} \tag{21}$$

using the notation $\mathbf{a} = [a_1, a_{1,x}, a_{1,y}, a_2, a_{2,x}, a_{2,y}]^{\mathrm{T}}$. Analogous to Sec. 2.2 we get the total differential

$$\frac{dI}{dt} = \partial_x I\dot{x} + \partial_y I\dot{y} + \partial_t I = I\dot{h}(\Delta X, \Delta Y, t, \mathbf{a}) \tag{22}$$

and eliminate optical flow $(\dot{x}, \dot{y})$ using (2) and (3)

$$\begin{aligned}\frac{\partial(I,Y)}{\partial(x,y)}U + \frac{\partial(X,I)}{\partial(x,y)}V + \frac{\partial(X,Y,I)}{\partial(x,y,t)} - Ia_1 - Ia_{1,x}\Delta X \\ - Ia_{1,y}\Delta Y - 2Ia_2 t - 2Ia_{2,x}\Delta X t - 2Ia_{2,y}\Delta Y t = 0\,.\end{aligned} \tag{23}$$

4 Parameter Estimation

For parameter estimation in a total least squares framework we closely follow [13]. The range constraint (see Sec. 2.1) yields for every pixel an equation of the form $\mathbf{d}_{rc}^{\mathrm{T}}\mathbf{p} = 0$ with

$$\mathbf{d}_{rc} = \left[\frac{\partial(Z,Y)}{\partial(x,y)}, \frac{\partial(X,Z)}{\partial(x,y)}, \frac{\partial(Y,X)}{\partial(x,y)}, \frac{\partial(X,Y,Z)}{\partial(x,y,t)}\right]^{\mathrm{T}} \tag{24}$$

and $\mathbf{p} = [U, V, W, 1]^{\mathrm{T}}$. To solve this equation containing three unknowns, we assume that within a local neighborhood Ω one parameter vector $\mathbf{p}$ solves all equations but for an error $\mathbf{e}$. Minimizing error $\mathbf{e}$ in weighted L_2-norm yields

$$||\mathbf{e}|| = \mathbf{p}^{\mathrm{T}}\mathbf{J}_{\mathbf{rc}}\mathbf{p} \stackrel{!}{=} \min \tag{25}$$

with structure tensor $\mathbf{J}_{\mathbf{rc}} = \mathbf{W} * (\mathbf{d}_{rc}\mathbf{d}_{rc}^{\mathrm{T}})$ and averaging filter $\mathbf{W}$ defining the neighborhood Ω. As described in Secs. 2.2 and 3.2 to 3.4 we have more than one constraint for the U and V component of the range flow. Analogous to the range constraint $\mathbf{d}_{rc}^{\mathrm{T}}\mathbf{p} = 0$ the intensity (9) and the gradient constancy constraints (15) and (16) may be formulated as $\mathbf{d}_Q^{\mathrm{T}}\mathbf{p} = 0$ using

$$\mathbf{d}_Q = \left[\frac{\partial(Q,Y)}{\partial(x,y)}, \frac{\partial(X,Q)}{\partial(x,y)}, 0, \frac{\partial(X,Y,Q)}{\partial(x,y,t)}\right]^{\mathrm{T}} \tag{26}$$

with $Q = \{I, I_x, I_y\}$ respectively. By inserting zeros into the appropriate places of data vector $\mathbf{d}$ we adapt all constraints to the same dimensions. The brightness change model presented in Sec. 3.4 contains motion and brightness change parameters. Therefore both data vectors for the range and the brightness change constraint have to be enlarged by zeros appropriately. Parameter vector $\mathbf{p}$ then becomes $\mathbf{p} = [U, V, W, 1, \mathbf{a}^{\mathrm{T}}]^{\mathrm{T}}$.

As in [13] we combine the different constraints yielding a combined structure tensor that is simply the weighted sum of the different tensors of the depth and the intensity channels

$$\mathbf{J} = \mathbf{J_{rc}} + \sum_{i=1}^{j} \beta_i \mathbf{J_i} \tag{27}$$

weighted with constants β_i and 3 possible choices for j, namely for $j = 1, 2$, or 3, depending on the number of brightness constraint equations used and corresponding to the constraints proposed in Secs. 2.2 and 3.2 to 3.4. Constants β_i may be used to account for different signal-to-noise-ratios of the structure tensors. Furthermore the data channels should be scaled to same mean and variance before they are combined.

As is well-known, this equation is minimized by the eigenvector $\mathbf{b}$ to the smallest eigenvalue of $\mathbf{J}$. Range flow is then given by

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \frac{1}{b_4} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} . \tag{28}$$

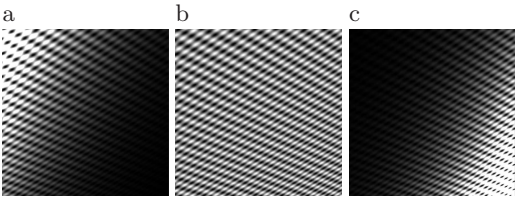


Fig. 2. Scaled first (a), central (b) and last (c) frame of sinusoidal sequence with illumination parameters $a_1 = 0$ and $a_{1,x} = 0.06$

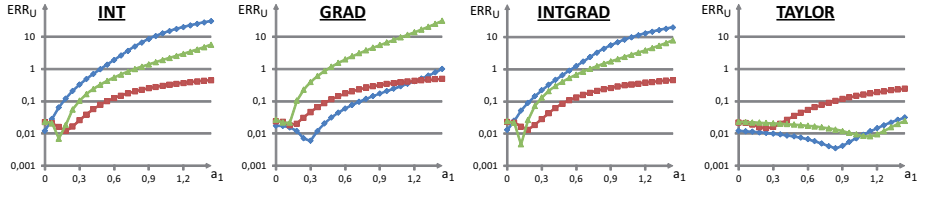


Fig. 3. Mean absolute value of relative error of U versus the brightness change parameter a_1 with no (-♦-), highpass (-▲-) and homomorphic (-■-) prefilters for different models

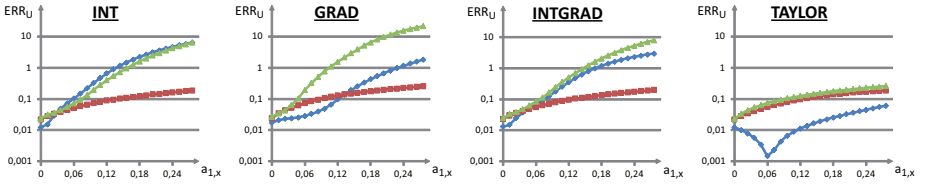


Fig. 4. Mean absolute value of relative error of U versus the brightness change parameter $a_{1,x}$ with no (-◆-), highpass (-▲-) and homomorphic (-■-) prefilters for different models

5 Experiments

For systematic error analysis of the different models sinusoidal patterns are used. The models compared are combinations of the range constraint with

1. the intensity constancy constraint (INT, (9)),
2. the gradient constancy constraint (GRAD, (15) and (16)),
3. the combined intensity and gradient constancy constraint (INTGRAD, (9), (15) and (16)) and
4. the intensity constraint with modeling of brightness changes by taylor series (TAYLOR, (23)).

Further accuracy of motion estimates on a rendered cube illuminated by a directed light source and motion estimation results on real data are demonstrated. For all experiments we generate range data by multi camera stereo reconstruction as presented in [28]. The weighting constants are set to $\beta_i = 1$ in all calculations.

5.1 Sinusoidal Pattern

For evaluation of the different models we modeled moving patches with sinusoidal patterns under varying illumination. Three frames of one test sequence are shown in Fig. 2. For our analysis we use patches translating with $U = 0.0073\,\text{mm/frame}$, $V = 0\,\text{mm/frame}$ and $W = 0.5\,\text{mm/frame}$ and angular velocity of $\omega = 0.002\,\text{radians/frame}$ around the Y-axis. For $t = 0$ the surface normal of the patch is $\mathbf{n} = (1, 2, -1)^{\mathrm{T}}$ and the distance of the patch center to the camera is $Z_0 = 100\,\text{mm}$. The synthetic sensor contains 301×301 pixels of size $0.0044\,\text{mm}^2$. Focal length of the synthetic projective camera is $f = 12\,\text{mm}$. For our analysis we compare the mean absolute value of the relative error of U

$$ERR_U = \frac{1}{N} \sum_{i}^{N} \frac{|U_{estimated} - U_{reference}|}{|U_{reference}|} \tag{29}$$

over all pixel N at a minimum distance of 60 pixel from the nearest image border. To reduce systematic errors the structure tensor weighting matrix $\mathbf{W}$ is realized by a 65-tab Gaussian with standard deviation $\sigma = 19$. Both prefilters

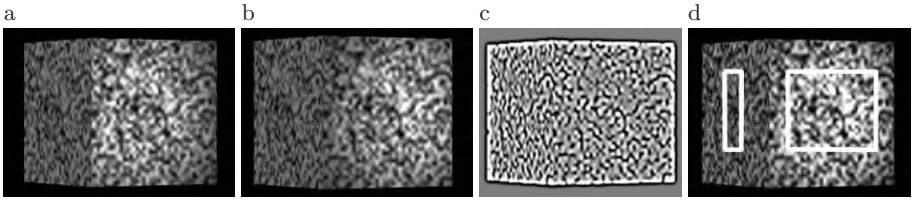


Fig. 5. First (a) and last (b) frame of cube sequence, (c) central frame of cube sequence after homomorphic prefiltering, (d) regions of cube used for error analysis

are generated as depicted in Sec. 3.1. Following [2] we compare the estimation of U for increasing illumination parameters $a_1|_{a_{1,x}=0}$ and $a_{1,x}|_{a_1=0}$ to simulate brightness changes. Only errors for U are presented, as errors of V and W showed similar characteristics.

In Fig. 3 and Fig. 4 errors of U for the proposed models in combination with prefilters are shown for increasing a_1 and $a_{1,x}$ respectively. Using homomorphic prefilters performances of all models are comparable. For models INT and INTGRAD homomorphic prefiltering is the best choice. For small brightness changes model GRAD with no prefilter produces even lower errors than using the homomorphic one. Lowest errors can be observed using model TAYLOR without prefilter. Overall these errors are about one order of magnitude smaller than for all other model and prefilter combinations over a wide range of brightness changes.

For models INT and INTGRAD the positive effect of reducing brightness changes using homomorphic prefiltering apparently overcompensates the negative effect of losing signal. For models GRAD and TAYLOR, which can handle brightness changes better, it seems to be the other way round. Reduced accuracy for non-changing brightness ($a_1 = 0$ and $a_{1,x} = 0$) proves the negative effect of loss of signal due to prefiltering.

5.2 Synthetic Cube

The synthetic cube sequence allows to test the models on more realistic data with groundtruth available. The cube moves with $U = -0.2\,\mathrm{mm/frame}$, $V = 0\,\mathrm{mm/frame}$ and $W = -2\,\mathrm{mm/frame}$. In addition to ambient light the cube is illuminated by a fixed light spot from the right. Figure 5 shows two frames of the cube sequence, a frame after homomorphic prefiltering as well as the investigated regions on the left and right side of the cube. The size of the weighting matrix $\mathbf{W}$ is realized by a 31-tab Gaussian with standard deviation $\sigma = 11$.

In Fig. 6 motion estimates of the standard range flow model INT is compared with the three models, which performed best on sinusoidal test sequences, i.e. model INTGRAD with homomorphic prefilter and both models GRAD and TAYLOR without prefilter. As errors are too small to be visible for all models the erroneous motion estimates are amplified for U and V by 50 and W by 20.

Table 1 shows numerical errors of the different models for the regions on the left and right side of the cube. We compare the average angular error [17]

$$AAE = \arccos\left(\frac{f_c f_e}{|f_c|\,|f_e|}\right)\,[^\circ] \tag{30}$$

and its standard deviation with the true and the estimated flow f_c and f_e respectively. Interested in estimating plant growth we furthermore compare the average relative growth rate and its standard deviation. According to [29] the relative change in size dA of a local surface area s parameterized in sensor coordinates x and y is calculated using the 3d displacement vector field f

$$dA = \frac{|s(x+1,y)+f(x+1,y) - (s(x,y)+f(x,y))| \times |s(x,y+1) + f(x,y+1) - (s(x,y)+f(x,y))|}{|s(x+1,y) - s(x,y)| \times |s(x,y+1) - s(x,y)|} \tag{31}$$

with $s(x,y) = [X(x,y), Y(x,y), Z(x,y)]^{\mathrm{T}}$. The relative growth rate is determined by $RGR = (dA-1)\cdot 100\%$. In the experiment the 3d structure of the cube remains constant in time, i.e. $RGR = 0\%$.

Additionally we show results obtained by the recent scene flow approach of [25]. We apply the algorithm the authors provide with the parameters of the *rotating sphere* experiment. Only the weighting parameter of the gradient constraint was increased to $\gamma = 30$ to reduce the effect of the severe brightness variations in the data.

Standard range flow estimation, i.e. model INT without prefilter (Fig. 6a) yields highly corrupted estimation results on the side of the cube where illumination changes due to the fixed spotlight. The other side does not suffer from illumination changes. There motion estimates are much more accurate. As expected estimates are well improved by the other models when brightness changes are present.

But using a homomorphic prefilter, e.g. with model INTGRAD (Fig. 6c), results on the left side of the cube are visibly worse than using standard range flow. Models without prefilter (Fig. 6b+d) or with a highpass yield more or less the same good results as standard range flow on the left side of the cube. These observations coincide with the errors in Table 1.

On its right side, where the brightness changes dominate, model TAYLOR (Fig. 6d) yields slightly more uniform, more accurate motion vectors than model GRAD (Fig. 6b). Moreover Table 1 shows that model TAYLOR is more robust with respect to prefiltering. With no prefilter or using a highpass rarely changes results, whereas the other models yield much more varying errors depending on the prefilter. The errors of [25] are much worse than any other used model, despite the elaborate estimator. We assume that tuning parameters may reduce errors a bit but did not find a good parameter set. This shows that using an elaborate estimator does not necessarily help, if the brightness model does not fit.

We conclude that homomorphic prefiltering should be avoided if possible and modeling brightness changes yields slightly more accurate results than using a model suppressing effects of brightness changes.

Table 1. Average angular error, average relative growth rate and their standard deviations of regions on left and right side of the cube (see Fig. 5d). High and low errors indicated in red and green respectively.

model	prefilter	left region AAE	left region RGR	right region AAE	right region RGR
INT	NO	0.148 ± 0.074	0.093 ± 0.092	4.311 ± 4.024	4.372 ± 9.211
	HP	0.137 ± 0.074	0.105 ± 0.097	0.069 ± 0.045	0.017 ± 0.057
	HOM	0.322 ± 0.216	0.087 ± 0.429	0.096 ± 0.148	-0.175 ± 0.564
GRAD	NO	0.142 ± 0.075	0.097 ± 0.094	0.120 ± 0.111	0.042 ± 0.102
	HP	0.135 ± 0.072	0.110 ± 0.098	0.046 ± 0.017	-0.003 ± 0.057
	HOM	0.664 ± 0.471	0.039 ± 0.862	0.139 ± 0.280	-0.29 ± 0.831
INTGRAD	NO	0.147 ± 0.074	0.094 ± 0.092	4.400 ± 4.081	4.500 ± 9.356
	HP	0.137 ± 0.073	0.107 ± 0.097	0.085 ± 0.066	0.028 ± 0.061
	HOM	0.240 ± 0.146	0.090 ± 0.316	0.084 ± 0.110	-0.14 ± 0.465
TAYLOR	NO	0.0149 ± 0.074	0.090 ± 0.094	0.055 ± 0.014	-0.009 ± 0.052
	HP	0.139 ± 0.074	0.105 ± 0.097	0.047 ± 0.016	-0.012 ± 0.051
	HOM	0.323 ± 0.216	0.087 ± 0.431	0.096 ± 0.149	-0.02 ± 0.565
[25]	NO	4.941 ± 1.812	-0.739 ± 11.59	2.216 ± 1.223	-3.36 ± 8.694
	HP	8.977 ± 3.714	-3.20 ± 15.51	4.041 ± 4.078	-4.19 ± 1.557
	HOM	8.907 ± 2.995	-1.83 ± 13.02	2.093 ± 0.784	0.347 ± 7.358

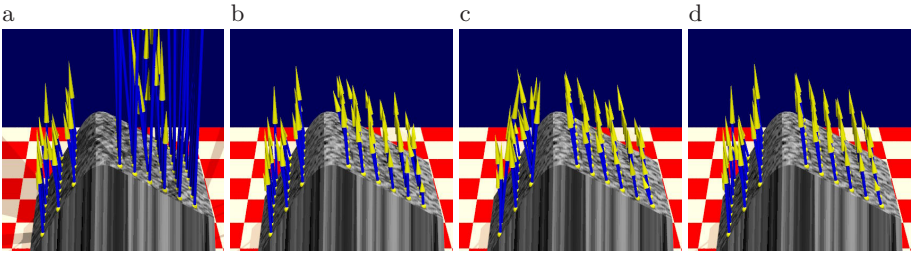


Fig. 6. Scaled motion estimates with amplified errors for different models: (a) INT without prefiltering, (b) GRAD without prefiltering, (c) INTGRAD with homomorphic prefiltering, (d) TAYLOR without prefiltering

5.3 Real Data

The previous experiments showed that model TAYLOR yields most reliable estimates. In Fig. 1 we show estimated 3d velocity fields for a freely moving castor bean plant leaf. The scene is illuminated by directed infrared light emitting diodes from the top right causing shadows on the leaf of interest. Depth reconstruction was obtained according to [28] using five images from different camera positions at each time step. For motion estimation a sequence of nine frames from the center camera with a sampling rate of 1 frame per 2 minutes was taken, i.e. acquisition time for the used images was 16 minutes.

The leaf rotates around the node where it is attached to the stem. This results in a visible motion towards the camera and to the right while the shadow area caused by the top leaf decreases. Model TAYLOR visibly improves estimation results in regions with illumination changes compared to the standard range flow model. As expected for a rigid motion, which can be assumed using a high temporal resolution we obtain a smoothly varying vector field.

6 Summary and Outlook

In this paper we extended range flow estimation presented in [13] with different approaches to handle inhomogeneous illumination. We presented a detailed error analysis for four different model constraints in combination with highpass and homomorphic prefilters on synthetic sequences with sinusoidal patterns and a translating cube. While prefilters improved estimation results on data when illumination changes are present, they suppress information producing worse results when no changes are present. Modeling brightness changes instead of prefiltering provides equal or better estimation results whether illumination changes occur or not.

In future work we plan to further improve the accuracy of the model and the estimator.

Acknowledgments. The authors would like to thank Georg Dreissen for his help with the acquisition of the plant leaf sequence.

References

1. Haußecker, H., Fleet, D.J.: Computing optical flow with physical models of brightness variation. PAMI 23, 661–673 (2001)
2. Schuchert, T., Scharr, H.: Simultaneous estimation of surface motion, depth and slopes under changing illumination. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) DAGM 2007. LNCS, vol. 4713, pp. 184–193. Springer, Heidelberg (2007)
3. Simoncelli, E.P.: Design of multi-dimensional derivative filters. In: ICIP (1), pp. 790–794 (1994)
4. Scharr, H.: Optimal filters for extended optical flow. In: International Workshop on Complex Motion, pp. 14–29 (2004)
5. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: DARPA Im. Underst. Workshop, pp. 121–130 (1981)
6. Bigün, J., Granlund, G., Wiklund, J.: Multidimensional orientation estimation with applications to texture analysis and optical flow. IEEE Trans. PAMI 13, 775–790 (1991)
7. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. IJCV 61, 211–231 (2005)
8. Black, M.J., Anandan, P.: The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. Comput. Vis. Image Underst. 63, 75–104 (1996)
9. Ju, S.X., Black, M.J., Jepson, A.D.: Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In: Proc. Computer Vision and Pattern Recognition, CVPR-1996, San Francisco, pp. 307–314 (1996)
10. Yamamoto, M., Boulanger, P., Beraldin, J.A., Rioux, M.: Direct estimation of range flow on deformable shape from a video rate range camera. IEEE PAMI 15, 82–89 (1993)
11. Gharavi, H., Gao, S.: 3-d motion estimation using range data. IEEE Transactions on intelligent transportation systems 8, 133–143 (2007)
12. Spies, H., Haußecker, H., Jähne, B., Barron, J.: Differential range flow estimation. In: DAGM, pp. 309–316 (1999)
13. Spies, H., Jähne, B., Barron, J.: Range flow estimation. CVIU 85, 209–231 (2002)

14. Zhang, Y., Kambhamettu, C.: Integrated 3d scene flow and structure recovery from multiview image sequences. In: CVPR, pp. 2674–2681 (2000)
15. Vedula, S., Baker, S., Rander, P., Collins, R., Kanade, T.: Three-dimensional scene flow. IEEE PAMI 27, 475–480 (2005)
16. Scharr, H.: Towards a multi-camera generalization of brightness constancy. In: Jähne, B., Mester, R., Barth, E., Scharr, H. (eds.) IWCM 2004. LNCS, vol. 3417, pp. 78–90. Springer, Heidelberg (2007)
17. Barron, J., Fleet, D., Beauchemin, S.: Performance of optical flow techniques. IJCV 12(1), 43–77 (1994)
18. Jähne, B., Haußecker, H., Geissler, P.: Handbook of Computer Vision and Applications, 1st edn. Academic Press, London (1999)
19. Bruhn, A.: Variational optic flow computation, accurate modelling and efficient numerics (2006)
20. Papenberg, N., Bruhn, A., Brox, T., Didas, S., Weickert, J.: Highly accurate optic flow computation with theoretically justified warping. IJCV 67, 141–158 (2006)
21. Denney, T.S.J., Prince, J.L.: Optimal brightness functions for optical flow estimation of deformable motion. IEEE Trans. Im. Proc. 3, 178–191 (1994)
22. Toth, D., Aach, T., Metzler, V.: Illumination-invariant change detection. In: SSIAI, pp. 3–7 (2000)
23. Oppenheim, A., Schafer, R., Stockham, T.G.: Nonlinear filtering of multiplied and convolved signals. In: Proceedings of the IEEE, vol. 56, pp. 1264–1291 (1968)
24. Pons, J.P., Keriven, R., Faugeras, O.: Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. IJCV 72(2), 179–193 (2007)
25. Huguet, F., Devernay, F.: A variational method for scene flow estimation from stereo sequences. In: ICCV (2007)
26. Li, R., Sclaroff, S.: Multi-scale 3d scene flow from binocular stereo sequences. wacv-motion 2, 147–153 (2005)
27. Jähne, B.: Digitale Bildverarbeitung, 4th edn. Springer, Heidelberg (1997)
28. Scharr, H., Schuchert, T.: Simultaneous motion, depth and slope estimation with a camera-grid. In: Vision, Modeling and Visualization 2006, pp. 81–88 (2006)
29. Spies, H., Jähne, B., Barron, J.L.: Surface expansion from range data sequences. In: Radig, B., Florczyk, S. (eds.) DAGM 2001. LNCS, vol. 2191, pp. 163–169. Springer, Heidelberg (2001)

Some Objects Are More Equal Than Others: Measuring and Predicting Importance

Merrielle Spain and Pietro Perona

California Institute of Technology
{spain,perona}@caltech.edu

Abstract. We observe that everyday images contain dozens of objects, and that humans, in describing these images, give different priority to these objects. We argue that a goal of visual recognition is, therefore, not only to detect and classify objects but also to associate with each a level of priority which we call 'importance'. We propose a definition of importance and show how this may be estimated reliably from data harvested from human observers. We conclude by showing that a first-order estimate of importance may be computed from a number of simple image region measurements and does not require access to image meaning.

1 Introduction

'Image understanding', the grand goal of machine vision, is about computing meaningful and informative semantic descriptions from images.

Progress in visual recognition has been breathtaking during the past 10 years. We now have algorithms that can recognize individual objects accurately and quickly [1], classify scenes [2], and learn new categories with little supervision [3,4,5,6,7].

What are the next steps toward image understanding? A full description of complex scenes, currently appears to be out of reach (although there is interesting work in that direction [8]). An intermediate goal is generating a list of keywords for each picture. This simpler description would be useful for indexing into large image databases (think of flickr.com's keyword system) and it would be readily understandable by humans. How should such a list be produced? As we shall see later, medium-resolution images of everyday scenes contain dozens of recognizable objects. A number of research groups are making quick progress on simultaneous recognition and segmentation [9,10,11]. However, rattling off an alphabetized list of nouns would not be particularly informative — not all objects are equal. So our goal is to produce a list of the *important* objects in the scene. We formalize the concept of importance as

> An object's *importance* in a particular image is the probability that it will be mentioned first by a viewer.

This paper is about defining, measuring, and predicting the importance of objects in images. Figure 1 depicts how our ideas fit together. Section 2 describes

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 523–536, 2008.

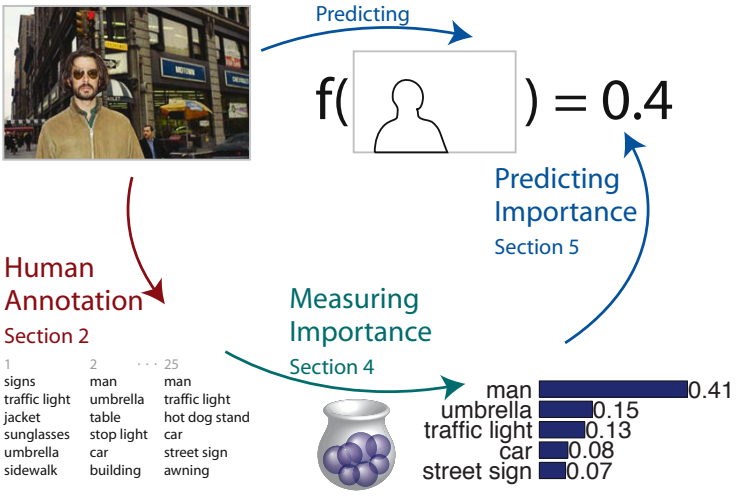


Fig. 1. Which objects matter in a scene? We can measure the importance of an object in a photo by combining lists of the objects named by different viewers (Section 2). To this end, we introduce an urn model, treating object naming as drawing balls from an urn (Section 4). Using these measurements we learn a function to predict object importance directly from photo regions (Section 5).

how we collect words from human observers. In Section 3 we explore how many objects there are in individual images and collections of images. Section 4 introduces a model for object naming based on importance. We show that this model accounts for both object naming frequency and order. This model, in turn, suggests a method for estimating importance from lists of objects produced by human observers. Section 5 explores whether object importance may by predicted directly from bottom-up visual properties of an object. We conclude in Section 6 with a discussion of our main findings.

2 Human Annotation

Intuitively, an important object is one that could help you identify or recreate the scene. In this section we describe how we collect data that enables measurement of importance.

2.1 Previous Work

The ESP game, by Ahn & Dabbish [12], presents the same image to two players who cannot communicate. Their task is to produce the same word in as few tries as possible. When the players produce the same word, the game ends, banning that word for future plays involving the same image and different players. It is intuitive that the word must be, in some way, related to the image. When multiple games are played on the same image, a list of words is produced, one

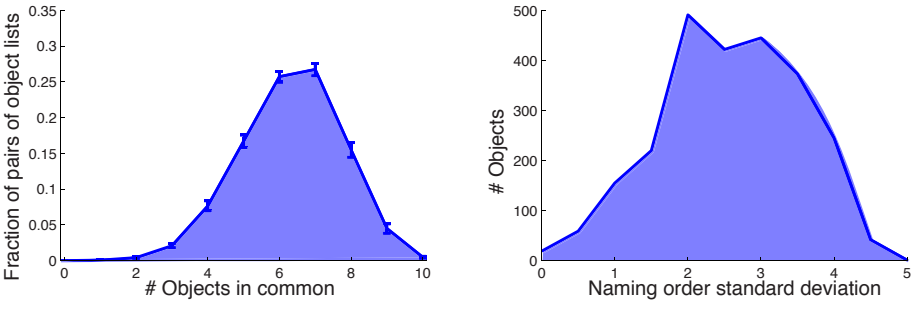


Fig. 2. Humans generate different lists of objects. When two humans independently name 10 objects in a single image, only two thirds of their lists tend to overlap *(left)*. It is extremely rare for the lists to contain all the same objects. The standard deviation in the order that humans name an object in a particular image is large *(right)*.

per game. The order of words in the list could measure importance. However, there are weakness in this approach: words are not always object names (e.g. funny) and word order is a noisy correlate of importance since only a pair of players need to produce a given word.

In LabelMe [13] users name objects and outline their contours with mouse clicks. A user may annotate as many objects as they like in an image. Results from previous users are visible to following users, so each object instance is annotated at most once. Elazary and Itti consider the object annotation order a measure of *interestingness* [14]; however, as partial annotations are passed on to new users, a single ordering is produced. Figure 2 shows that when you compare lists of objects named by several humans for the same photo, different objects are named and the object naming order varies wildly. Hence a single list will not capture the statistics of object naming.

2.2 Our Approach

We needed an unbiased, representative, and meaningful set of scenes for our experiments. By 'unbiased', we mean that that the choice of scenes should be as independent as possible from the experimenters and the purpose of the experiment. By 'representative', we mean that the collection of images should sample human visual experience as broadly as possible. However, if the images had been collected completely at random, that is by attaching the camera to a someone's head and snapping one picture per minute for a day [15], most pictures would turn out to be uninterpretable and irrelevant; by 'meaningful', we mean that the images should represent meaningful moments in a person's visual experience. We selected our gallery of 97 pictures from Stephen Shore's collections 'American Surfaces' and 'Uncommon Places' [16,17]. Shore took these pictures while traveling in North America in the 70's and 80's and were meant to be a visual diary of his experience. Figures 3 and 4 show sample images.

Through the Amazon Mechanical Turk, observers (English speakers in the U.S.) named 10 objects that they saw in a scene photograph. Each photograph was displayed with a 600 pixel diagonal and annotated by 25 different observers. While previous approaches produce a single word list, we have 25 ordered lists for each image–hence we can use statistical regularities to quantify the importance, not just order of objects.

We used WordNet [18] primary definitions to map synonyms and plurals to the same word, and match missed synonyms (such as misspellings) by hand. As Figure 2 shows, the objects that a viewer names and the order in which a viewer names them vary wildly. Figures 3 and 4 show median order vs. naming frequency (across viewers) for our sample images. Each point corresponds to an object; if an object is mentioned by 35% of the viewers, it has a .35 x-coordinate. The y-coordinate represents the median naming order of the object. So if three observers name a particular object, and it is named 1st, 4th, and 10th, then the y-coordinate is 4. Section 4 introduces a simple model, the urn model, that generates object sequences from object importances, enabling us to measure object importances from human generated sequences.

3 Object Counts

A first observation we make in assessing our data, is that there are many objects named in each picture (see Figure 5). For each image, some of these objects are mentioned by a few observers, while other objects are mentioned by many observers. The heights of the curves in Figure 5 quantify how rich with objects images, environments, and the world (full collection) are. The dashed line shows that many fewer objects are named by at least 5 people. The difference between the solid and dashed lines is the number of objects that are rarely named, in that 80% of people don't name them. Hence Figure 5 shows that there are many recognizable objects in a scene photograph, but few objects are preferred by viewers. Hence we need to quantify how important an object is, in order to describe a scene meaningfully with the objects that it contains.

4 Measuring Importance

As proposed in the introduction, we define an object's *importance* in a photo as the probability that a human observer naming objects will name it first. In principle, we would need an extraordinary number of observers to be able to directly calculate the importance of all the objects in a picture: some objects' importances may be less than 1%, and we would need hundreds of observers to determine that. In this section we show that it is possible to measure an objects' importance from relatively few observers if we are willing to model the process that generates an observer's sequence.

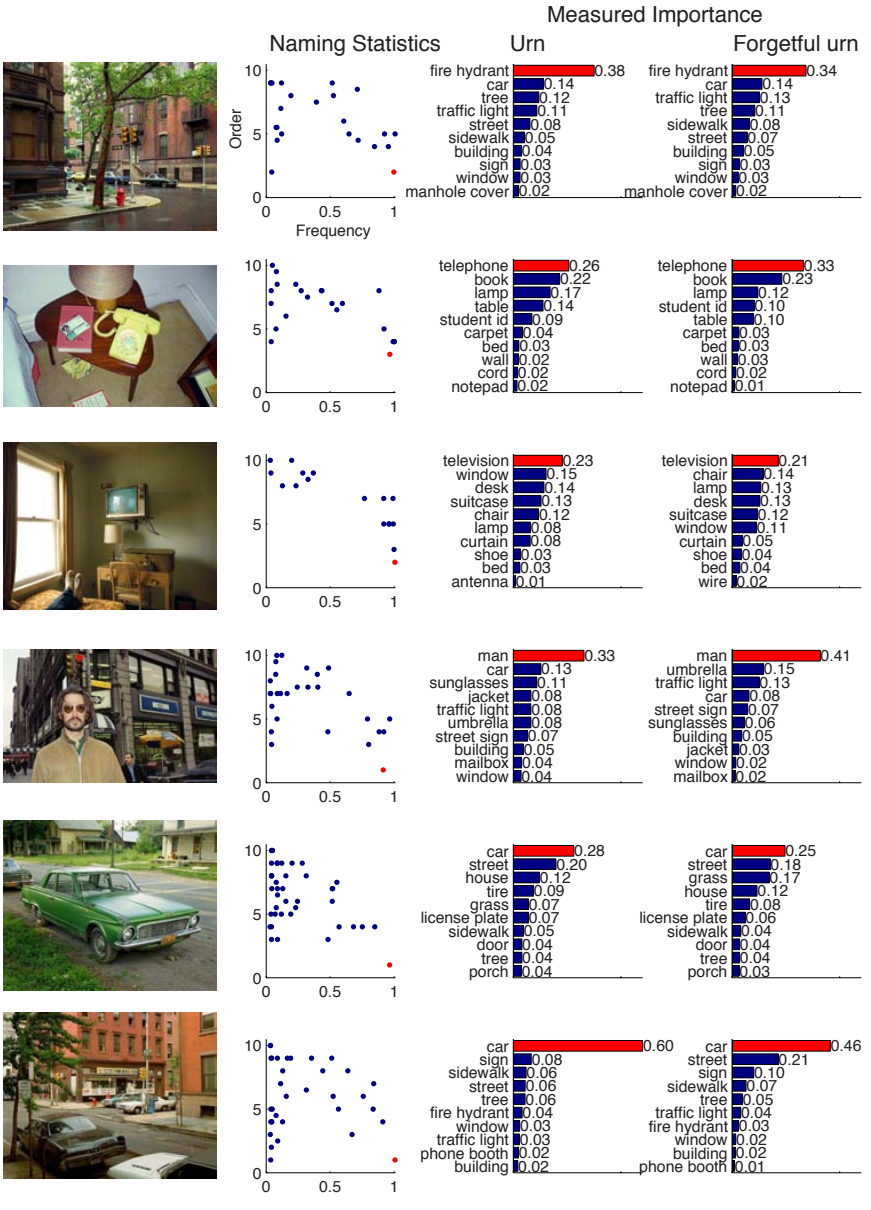


Fig. 3. Examples where the urn model fits the data well. An object's *(dot's)* median order named and frequency mentioned *(2nd column)* are in agreement given the model. In these cases, the forgetful urn *(4th column)* produces similar measured importance to the urn model *(3rd column)*.

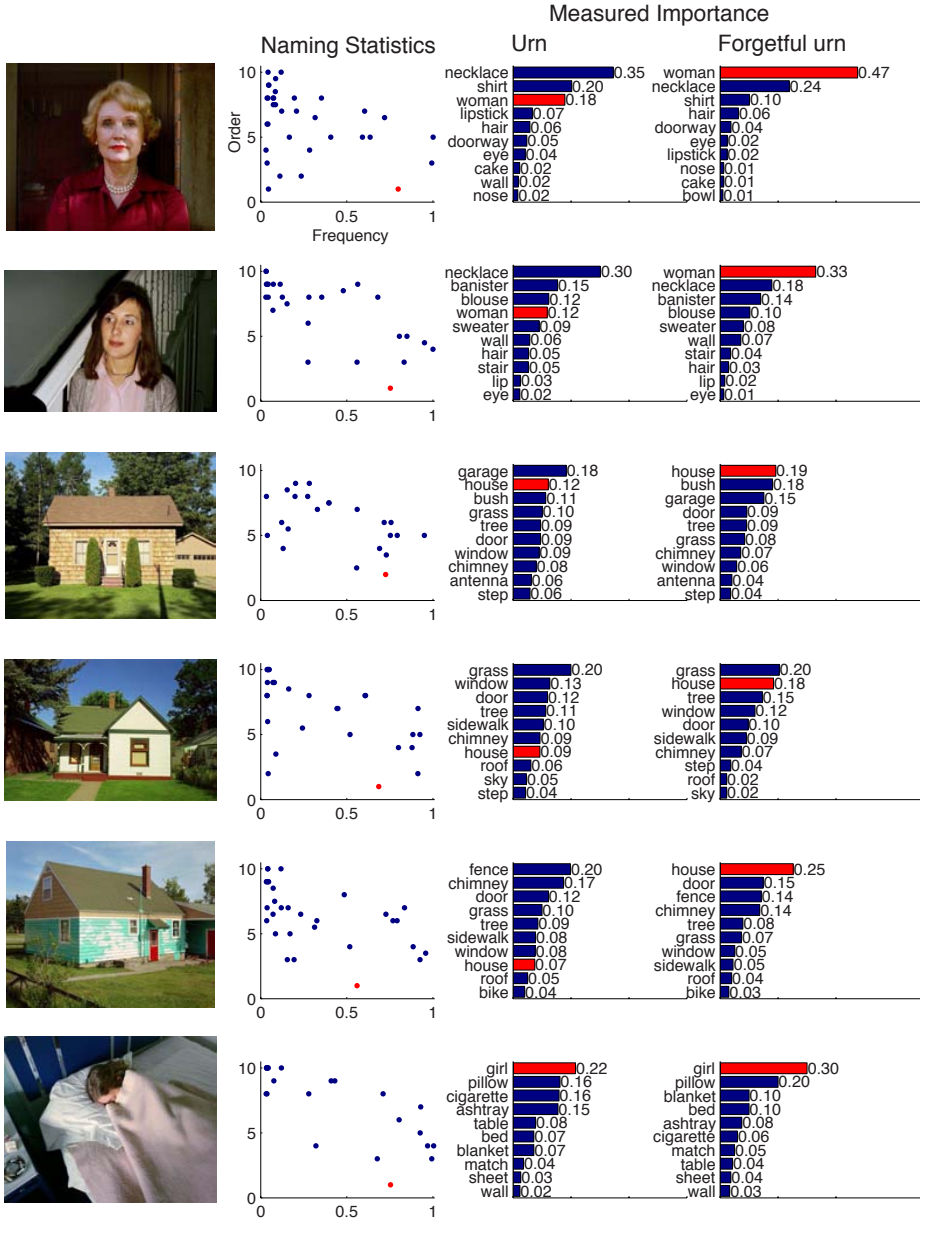


Fig. 4. In many photos with a strong central object, the urn model fails to identify the most important object. The poor behavior of the urn model *(3rd column)* is due to the fact that some viewers fail to name the central object *(red dot)*, while the viewers that name it, name it early *(2nd column)*. We propose a modified model, the forgetful urn *(4th column)* to solve this problem.

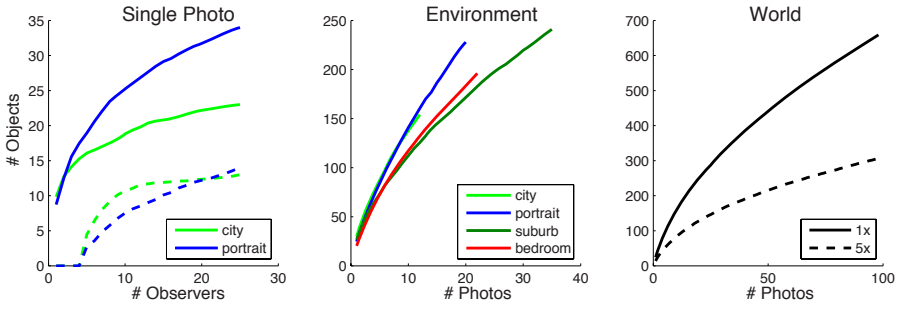


Fig. 5. The number of unique objects named in a photo or collection of photos is surprisingly large and most of these objects are rarely mentioned. The number of unique objects *(solid lines)* named in a photo (top in Figures 3 and 4) as observers are added *(left)*, in an environment as photos are added *(middle)*, and in the full collection as photos are added *(right)*. Many fewer objects have been named by at least 5 viewers *(dashed lines)*. This *(height of the solid line)* displays the sheer number of objects and the large proportion *(solid line - dashed line)* of rarely named objects.

4.1 Urn Model

We model the process of naming objects in an image with the process of drawing balls from an urn without replacement (see Figure 6). The balls are different sizes, affecting their probability of being chosen first. Thus, a ball's size represents the importance of the corresponding object. We represent multiple players by repeatedly refilling the urn with the same set of balls and selecting new sequences independently of each other.

Figure 7 shows that randomly drawn lists (of balls) from the urn model can reproduce, at least qualitatively, the order and frequency found in human-generated lists of object names. The synthetic data does not actually correspond to objects in images, but to abstract balls in the urn model. This is a phenomenological model with matches our observations for many images. In order to estimate the size of the balls (the importance of each object) from our human data, we follow the inverse process, that is starting from lists of objects one computes the Maximum Likelihood (ML) values of the parameters of the model (Figures 3 and 4). However, we note that for many photographs composed with a central object, 10-30% of viewers fail to mention that object, (see Figure 4). In the next section we describe our ML method and provide a solution to this problem.

4.2 Fitting the Urn

In order to estimate importance, we could count how often an object is named first, however, this squanders naming order information. When we have limited data (human annotations), finding the Maximum Likelihood Estimator (MLE) of urn model parameters improves upon the direct calculation of importance.

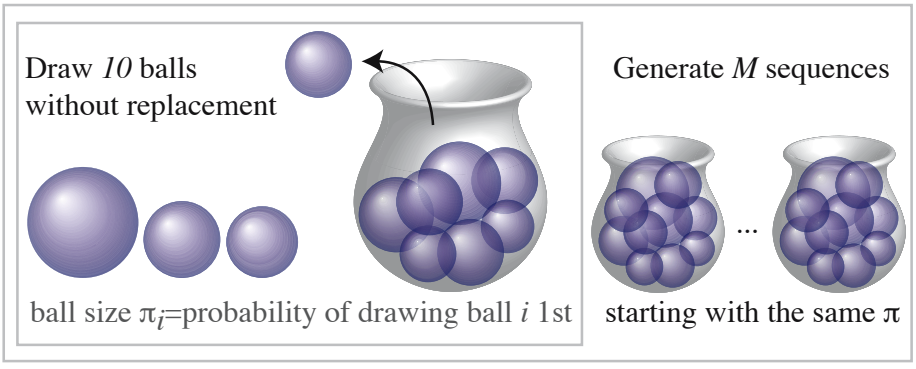


Fig. 6. Urn model relates object importance to named sequences of objects. An urn (image) is filled with balls (objects), having probabilities π_i of being drawn first (importances). 10 balls are drawn (named) from the urn without replacement, creating a sequence. M sequences are drawn.

This is a special case of estimating the weights from a Multivariate Wallenius' Noncentral Hypergeometric Distribution, which requires numerical methods [19]. Manly showed that the weights can be estimated if there are many (>10) balls with the same label being drawn [20], but we have only one ball with each label, so we cannot use his approach. To measure importance via the urn model we need to calculate the probability of observing a set of sequences given the object importances π_i.

Each sequence consists of 10 balls w_i^m (ith ball in the mth sequence) drawn independently without replacement (out of N balls, where $N >> 10$), so the probability of drawing a particular sequence of balls $(w_1^m, ...w_{10}^m)$ is

$$\prod_{n=1}^{10} p(w_n^m | w_{n-1}^m, ...w_1^m) \ . \tag{1}$$

However, we are drawing balls without replacement, so this equation is subject to the condition $w_i^m = w_j^m \implies i = j$. When we draw the nth ball of a sequence, $n-1$ balls have been removed from the urn. The probability that the ball labeled w_n^m is the nth ball drawn is therefore

$$p(w_n^m | w_{n-1}^m, ...w_1^m) = \begin{cases} 0 & \text{if } \exists i \in [1, n-1] : w_i^m = w_n^m, \\ \frac{\pi_{w_n^m}}{1 - \sum_{i=1}^{n-1} \pi_{w_i^m}} & \text{otherwise,} \end{cases} \tag{2}$$

where π_i is the probability that ball i is drawn first (from a fresh urn) and $\sum_i \pi_i = 1$. However, as Figure 4 shows, sometimes viewers fail to mention the most meaningful object. We believe that, when an object is very obvious, some subjects may quickly move beyond it without mention. We treat this as the first

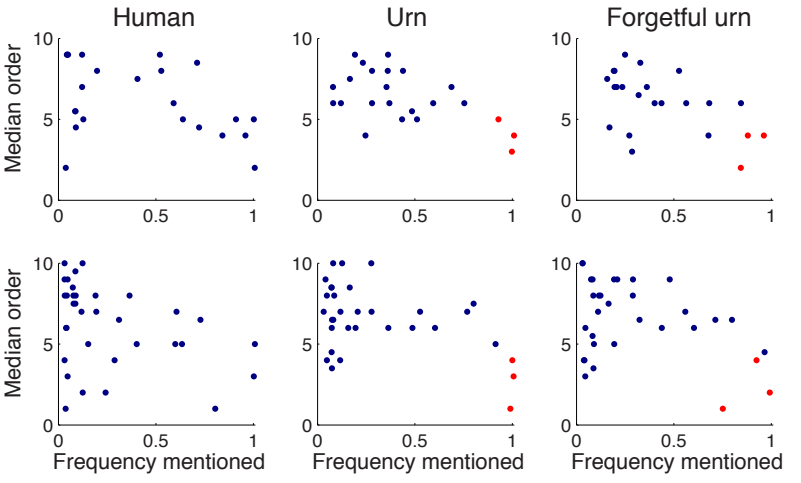


Fig. 7. The urn model reproduces peculiar characteristics of an object's *(dot's)* median order vs. frequency mentioned. Humans *(left)* and the urn model *(middle & right)* produce similar plots. The forgetful urn reproduces how, in some photos, viewers name the most important object either early or never, as if sometimes discarding the 1st ball they drew *(bottom right)*. In synthetic data, the 3 most important objects *(red dots)* are named early and often.

ball being drawn and discarded, and hence excluded from the sequence.[1] We deal with this strange problem in a non-standard way. Consider a sequence of balls where the first ball has been discarded (i.e. really drawn 1st, but not listed); the ball is most likely $\text{argmax}_{j:\forall_i j \neq w_i^m} \pi_j$, that is the most probable of the balls which haven't been mentioned. In this case π_j will likely be large. For a sequence of 10 balls in which the first ball was not dropped, π_j will probably be small. Hence, if we include $\max_{\forall_i j \neq w_i^m} \pi_j$ in the normalization, we obtain estimates of π_i that are not far from the correct ones when the first ball is not dropped and most often the correct estimates when the first ball is dropped. This changes Equation 2 to

$$p(w_n^m | w_{n-1}^m, ...w_1) = \frac{\pi_{w_n^m}}{1 - \max_{\forall_i j \neq w_i^m} \pi_j - \sum_{i=1}^{n-1} \pi_{w_i^m}} . \tag{3}$$

Since we have M independent sequences, the likelihood of our observation is

$$p(obs) = \prod_{m=1}^{M} \prod_{n=1}^{10} \frac{\pi_{w_n^m}}{1 - \max_{\forall_i j \neq w_i^m} \pi_j - \sum_{i=1}^{n-1} \pi_{w_i^m}} . \tag{4}$$

[1] So the rigorous definition of importance is the probability that a ball is drawn first, regardless of whether it is discarded.

To measure importance $\pi_{w_i^m}$, we maximize the log-likelihood $log(p(obs))$,

$$\sum_{m=1}^{M}\sum_{n=1}^{10}\log \pi_{w_n^m} - log(1 - \max_{\forall_i j \neq w_i^m} \pi_j - \sum_{i=1}^{n-1} \pi_{w_i^m}) \; . \tag{5}$$

Using synthetic data generated by an urn model Monte Carlo with different parameter settings, we see that the MLE of Equation 5 enables us to estimate the parameters more precisely than direct calculation from the observed frequency. Particularly, Figure 8 shows that the urn model and the forgetful urn model (the model containing the $\max_{\forall_i j \neq w_i^m} \pi_j$ term) are much closer to the true importance distribution in the K-L Divergence, $D_{KL}(\pi||\hat{\pi})$ [21]. The forgetful urn outperforms the original urn model when the first ball has a nonzero probability of being dropped, and is equivalent when the first ball isn't dropped. These are the mean K-L Divergence values over 5 importance distributions, each with 10 sets of 25 sequences drawn, shown at 4 probabilities of dropping the first ball.

Figures 3 and 4 display the importances of the 10 most important objects in our sample images, using the urn MLE and forgetful urn MLE.

One could wonder if our definition of importance captures objects that may never be named first. For instance in a photo of Batman and Robin, Robin may never be named first, yet he is important. In this example Robin's consistently second position violates the independence assumption of our model. The fitting will then assign a high importance to Robin. Empirically, we can take data from the urn model and move the second most important ball to second place every time it is drawn first. In our simulations this change does not significantly decrease the estimated importance of this ball (Wilcoxon Rank Sum Test).

Optimization Note: There are as many parameters π_i as objects mentioned. Figure 5 shows that this number can get large, which results in poor convergence. However if we limit our optimization to the 10 most frequently named objects and set the others to a small constant, our convergence using fmincon (repeatedly and perturbing the solutions) in the Matlab Optimization Toolbox is quite good.

5 Predicting Importance

Is it possible to automatically predict importance from image measurements without using human observers? A number of researchers are working on the problem of segmentation and recognition. In this paper we wish to focus on the orthogonal problem of estimating importance once the objects have been detected and segmented. Since no such system works sufficiently well nowadays, we segment the image by hand.

Using a training set of 30 images (687 objects), a validation set of 10 images (217 objects), and a test set of 35 images (800 objects), we fit a generalized linear model $log(\pi_i) = \sum_j X_{ij} b_j$. Here π_i is the measured importance of object i (from Section 4) in training and the predicted importance in testing. Our features consider the composition of the photo in relation to an object's outline; X_{ij} is object i's value for feature type j and b_j is the weight of the jth feature.

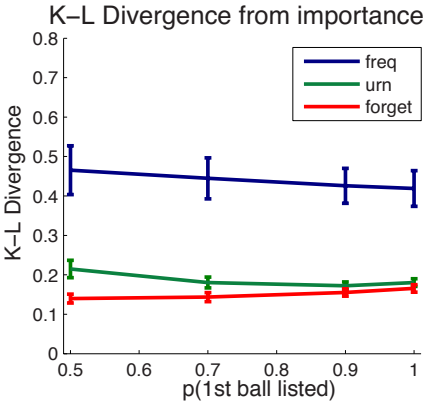


Fig. 8. The urn models measures importance more precisely than directly calculating how often an object is named first. An urn model Monte Carlo simulation generated sequences (with different probabilities of listing the first ball), and we used these sequences to estimate the synthetic parameters. The urn models' MLEs *(green, red)* are closer to the true parameters than the first mentioned frequency estimate *(blue)*. The forgetful urn model *(red)* is better than the plain urn model *(green)* when the first ball could be discarded, and equivalent otherwise.

As there were many more unimportant objects than important ones (importance > 0.07), we selected an equal number of each, using outlier removal. Our outlier removal was inspired by Angelova et. al's [22] idea that noisy data should be predicted poorly by a partial model that it didn't influence. We built many partial models and predicted the low importance training objects with them; we discarded the objects that gave the largest squared error across partial models.

We added features to each regression greedily, by choosing the feature that most reduced training error. We grew a regression starting at each feature, stopping growth at the lowest validation error. Finally, we selected the regression with the lowest training error. Out of a list of 30 features, the chosen features (in order of choice) were: distance from the image center to the closest part of the object, minimum distance from the object to the 4 points that divide the image into thirds (taken from the thirds rule of photographic composition), minimum vertical distance above the image center, maximum saliency[23] on the object, mean number of overlapping objects, total saliency over object, total blurred saliency over object (2 pixel blur on saliency map), maximum vertical distance below the image center, and maximum horizontal distance from the image center.

Our regression predicted importance (from photo patches) which we compared to measured importance (from human data in Section 4). We evalated the quality of such predictions quantitatively by building a binary classifier: objects were classified as as having higher or lower importance than a given importance threshold. Figure 9 shows the ROC curves at 3 importance levels, illustrating that our prediction method can discriminate very important objects. If we want

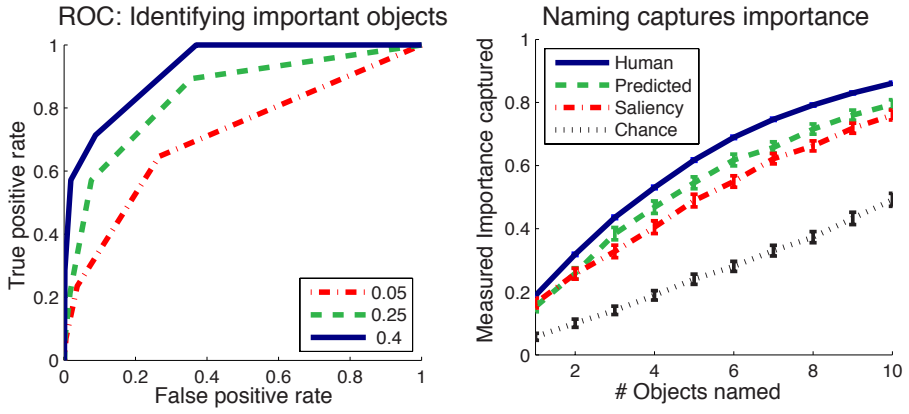


Fig. 9. Regression on image region properties can predict object importance. We define 'important' objects as having a measured importance > 0.05 *(red)*, 0.15 *(green)*, 0.4 *(blue)* and produce an ROC Curve for each definition *(left)*. We calculate how much measured importance has been captured by the first n objects named *(right)*. Humans *(blue)* outperform predicted importance *(green)*, which in turn outperforms the total saliency of an object *(red)*, which is much better than chance *(black)*.

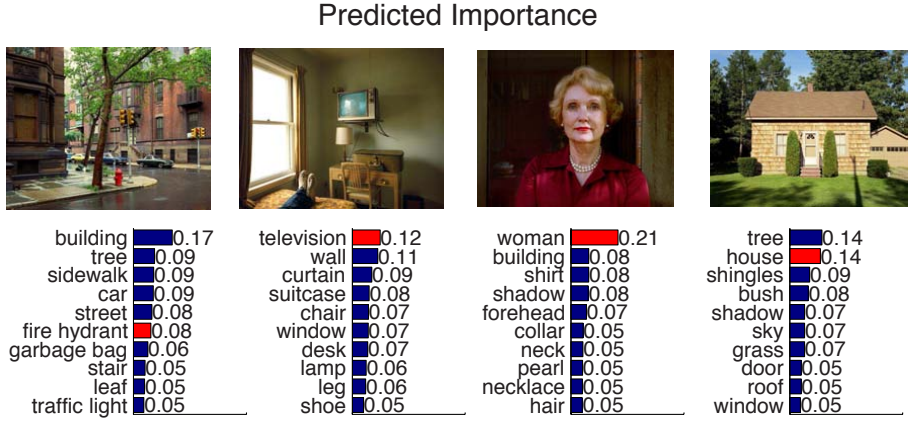


Fig. 10. Examples of predicted importances for sample images. The object with the highest measured importance *(red)* tends to have a high predicted importance. We train on image region properties with measured importance (Figures 3 and 4) as the desired output. Our testing output is predicted importance.

to name objects to capture as much measured importance as possible, we can sum the measured importance of the first n objects named. The naming order is obtained by sorting objects by predicted importance (or total saliency). Figure 9 shows the performance of predicted importance is sandwiched between human naming and saliency; it also shows that all 3 greatly outperform chance.

For a more intuitive evaluation, Figure 10 shows the importance predictions for sample photographs, the most noticeable discrepancy between automatically predicted importance (Figure 10) and importance estimated directly from human subjects (Figures 3 and 4) is that the fire hydrant was more important to people than our regression predicted.

6 Discussion

We asked a number of human observers to list the objects that they saw in images of complex everyday scenes; each image (of 97) was annotated by 25 observers. The data we collected shows that our visual world is rich (Figure 5): there are dozens of things that may be recognized in each image.

A number of research groups are making progress on simultaneous recognition and segmentation. Here we study the complementary problem of importance. Not all objects are equal: in a given image some are mentioned by all observers and some by few, some are mentioned early and some later. This suggests that we should not be content for our recognition algorithms to return a laundry list of things that are present in the image. We suggest that a complete system will require both recognition-segmentation and importance.

We defined 'importance' as the probability that an object is mentioned first by a human observer. We provided a methodology for measuring importance of objects in images from the data provided by our human observers. We noticed that human observers sometimes miss the most obvious object. We proposed a procedure to measure importance that takes this phenomenon into account. We found experimentally that our measurements of importance are reliable on synthetic data (Figure 8) and intuitive on human data (Figures 3 and 4).

One could worry that it may be impossible to assess an object's 'importance' automatically until the meaning of an image is understood [24]. We explored how far can one go in estimating object importance automatically from bottom-up image measurements. We found that a small number of simple measurements go a long way towards predicting importance (Figure 9).

Finally, it should be pointed out that this work is about photographs taken by humans — our findings may not generalize to haphazardly captured photographs.

Acknowledgments. This material is based upon work supported under a National Science Foundation Graduate Research Fellowship, National Institute of Mental Health grant T32MH019138, Office of Naval Research grant N00014-06-1-0734, and National Institutes of Health grant R01 DA022777. We would like to thank Antonio Torralba for insightful discussions and Ryan Gomes and Kristin Branson for useful corrections.

References

1. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (2004)
2. Oliva, A., Torralba, A.B.: Scene-centered description from spatial envelope properties. In: Biologically Motivated Computer Vision, pp. 263–272 (2002)

3. Weber, M., Welling, M., Perona, P.: Unsupervised learning of models for recognition. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 18–32. Springer, Heidelberg (2000)
4. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR, vol. 2, pp. 264–271 (2003)
5. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering objects and their localization in images. In: ICCV, 370–377 (2005)
6. Grauman, K., Darrell, T.: Efficient image matching with distributions of local invariant features. In: CVPR, vol. 2, pp. 627–634 (2005)
7. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR, vol. 2, pp. 2169–2178 (2006)
8. Barnard, K., Forsyth, D.A.: Learning the semantics of words and pictures. In: ICCV, pp. 408–415 (2001)
9. Russell, B.C., Efros, A.A., Sivic, J., Freeman, W.T., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: Proceedings of CVPR (2006)
10. Andreetto, M., Zelnik-Manor, L., Perona, P.: Unsupervised learning of categorical segments in image collections. In: Computer Vision and Pattern Recognition (CVPR 2008) (2008)
11. Todorovic, S., Ahuja, N.: Extracting texels in 2.5d natural textures. In: Proceeddings of ICCV (2007)
12. von Ahn, L., Dabbish, L.: Labeling images with a computer game. In: CHI, pp. 319–326 (2004)
13. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: a database and web-based tool for image annotation. Technical report (2005)
14. Elazary, L., Itti, L.: Interesting objects are visually salient. Journal of Vision 8, 1–15 (2008)
15. Mayer, M., Switkes, E.: Spatial frequency taxonomy of the visual environment. Investigative Ophthalmology and Visual Science 26 (1985)
16. Shore, S.: Stephen Shore: American Surfaces. Phaidon Press (2005)
17. Shore, S., Tillman, L., Schmidt-Wulffen, S.: Uncommon Places: The Complete Works. Aperture (2005)
18. (Wordnet)
19. Fog, A.: Calculation methods for wallenius' noncentral hypergeometric distribution. Communications In statictics, Simulation and Computation 37, 258–273 (2008)
20. Manly, B.F.J.: A model for certain types of selection experiments. Biometrics 30, 281–294 (1974)
21. Kullback, S., Leibler, R.A.: On information and sufficiency. Annals of Mathematical Statistics 22, 79–86 (1951)
22. Angelova, A., Matthies, L., Helmick, D.M., Perona, P.: Fast terrain classification using variable-length representation for autonomous navigation. In: CVPR (2007)
23. Walther, D., Koch, C.: Modeling attention to salient proto-objects. Neural Networks 19, 1395–1407 (2006)
24. Yarbus, A.: Eye movements and vision. Plenum Press, New York (1967)

Robust Multiple Structures Estimation with J-Linkage

Roberto Toldo and Andrea Fusiello

Dipartimento di Informatica, Università di Verona
Strada Le Grazie 15, 37134 Verona, Italy
andrea.fusiello@univr.it

Abstract. This paper tackles the problem of fitting multiple instances of a model to data corrupted by noise and outliers. The proposed solution is based on random sampling and conceptual data representation. Each point is represented with the characteristic function of the set of random models that fit the point. A tailored agglomerative clustering, called J-linkage, is used to group points belonging to the same model. The method does not require prior specification of the number of models, nor it necessitate parameters tuning. Experimental results demonstrate the superior performances of the algorithm.

1 Introduction

A widespread problem in Computer Vision is fitting a model to noisy data: The RANSAC algorithm is the common practice for that task. It works reliably when data contains measurements from a single structure corrupted by gross outliers.

When multiple instances of the same structure are present in the data, the problem becomes tough, as the robust estimator must tolerate both gross outliers and *pseudo-outliers*. The latter are defined in [1] as "outliers to the structure of interest but inliers to a different structure". The difficulty arise because robust estimators, including RANSAC, are designed to extract a single model. Sequential RANSAC – sequentially apply RANSAC and remove the inliers from the data set as each model instance is detected – has been proposed as a solution, but [2] argued that the approach is non optimal, and introduced the multi-RANSAC algorithm. The method is effective (provided that the models do not intersect each other), but the number of models is user specified, and this is not acceptable in some applications.

Another popular method based on random sampling and voting is the Randomized Hough Transform (RHT) [3]. It builds an histogram over the parameter space. A minimal sample set is randomly selected and the parameters of the unique model that it defines are recorded in the histogram. Peaks in the histogram correspond to the sought model. Its extension to multiple models is straightforward: they are revealed as multiple peaks in the parameter space. Hence, RHT does not need to know the number of models beforehand. However,

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 537–547, 2008.

RHT suffers from the typical shortcomings of Hough Transform methods, such as limited accuracy and low computational efficiency. Ultimately, the choice and the discretization of the parameter space turn out to be crucial.

RHT can be seen as an instance of a more general approach consisting of finding modes in parameter space (see e.g. [4]). Instead of quantize the parameter space and accumulate votes, one can map data into the parameter space through random sampling and then seek the modes of the distribution with mean-shift [5]. This, however, is not an intrinsically robust technique, even if it can be robustified with outliers rejection heuristics. Moreover, the choice of the parametrization is critical, as in RHT.

In summary, RANSAC is very robust, but it is not suited to deal with multiple structures. Mode finding in parameter space (and RHT), on the contrary, copes naturally with multiple structures, but cannot deal with high percentage of gross outliers, especially as the number of models grows and the distribution of inliers per model is uneven. Also the algebraic technique presented in [6] is effective in estimating multiple models, but it is not robust to gross outliers.

Recently [7] proposed a novel method for estimating multiple structures based on the analysis of the distribution of residuals of individual data points with respect to the hypotheses, generated by a RANSAC-like sampling process. It turns out that the modes of the residuals distribution reflects the model instances. With respect to RANSAC and other robust methods (such as LMeds, for example) this entails a change of perspective: "studying the distribution of the residuals for each data point instead of studying the distribution of residuals per each hypothesis" [7].

Residuals for each data point have peaks corresponding to the true models because hypotheses generated with random sampling tend to cluster around the true model, a fact that is also at the basis of RHT. The method, in principle, can discover the number of models automatically as in RHT and is effective as RANSAC. However, finding modes ends up to be cumbersome, as proved in our experiments. One reason is that the peak corresponding to a given model becomes less localized as the point-model distance increases. As a result, the rightmost modes in the histogram are usually drowned in the noise.

In the same spirit of RHT and [7] we exploit clustering of the hypotheses. However we do not work in the parameter space, which is at the root of the shortcoming of Hough Transform, nor in the residual space, which leads to the difficulties of modes estimation [7]. We adopt instead a *conceptual representation*: each data point is represented with the characteristic function of the set of models preferred by that point[1]. Multiple models are revealed as clusters in the conceptual space. Experimental comparison with sequential RANSAC, multiRANSAC, residual histogram analysis [7] and mean-shift is favorable to our algorithm.

[1] According to [8] the posterior probabilities of an object x given C classes form a *similarity conceptual representation*: $[P(x|\text{ class } 1) \cdots P(x|\text{ class } C)]$.

2 Method

The method starts with random sampling: M model hypothesis are generated by drawing M minimal sets of data points necessary to estimate the model, called minimal sample sets (MSS). Then the consensus set (CS) of each model is computed, as in RANSAC. The CS of a model is the set of points such that their distance from the model is less than a threshold ε.

Imagine to build a $N \times M$ matrix where entry (i, j) is 1 if point i belongs to the CS of model j, 0 otherwise. Each column of that matrix is the characteristic function of the CS of a model hypothesis. Each row indicates which models a points has given consensus to, i.e., which models it prefers. We call this the *preference set* (PS) of a point. Figure 1 shows an example of such a matrix in a concrete case.

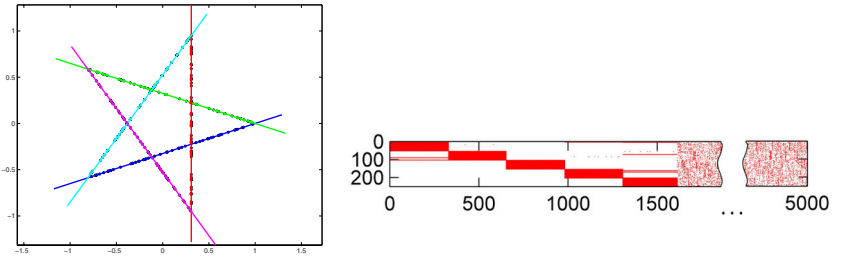


Fig. 1. Right: the data consist of 250 points on five segments forming a star. Left: Preference matrix. The rows are points (ordered by cluster), the columns are models (ordered by cluster size).

The characteristic function of the preference set of a point can be regarded as a conceptual representation of that point. Points belonging to the same structure will have similar conceptual representations, in other words, they will cluster in the *conceptual space* $\{0, 1\}^M$. This is, again, a consequence of the fact that models generated with random sampling cluster in the hypothesis space around the true models.

2.1 Random Sampling

Minimal sample sets are constructed in a way that neighbouring points are selected with higher probability, as suggested in [9,2]. Namely, if a point $\mathbf{x}_i$ has already been selected, then $\mathbf{x}_j$ has the following probability of being drawn:

$$P(\mathbf{x}_j|\mathbf{x}_i) = \begin{cases} \frac{1}{Z} \exp -\frac{||\mathbf{x}_j - \mathbf{x}_i||^2}{\sigma^2} & \text{if } \mathbf{x}_j \neq \mathbf{x}_i \\ 0 & \text{if } \mathbf{x}_j = \mathbf{x}_i \end{cases} \tag{1}$$

where Z is a normalization constant and σ is chosen heuristically.

Then for each points its preference set is computed, as the set of models such that the distance from the point is less than the inlier threshold ε (same as RANSAC).

The number M of MSS to be drawn is related to the percentage of outlier and must be large enough so that a certain number (at least) of outlier-free MSS are obtained with a given probability for all the models. Please note that if this condition is verified for the model with less inliers, it is automatically verified for all the other models.

Let S be the number of inliers for a given model and N be the total number of points. The probability of drawing a MSS of cardinality d composed only of inliers is given by:

$$p = P(E_1)P(E_2|E_1)\dots P(E_d|E_1, E_2\dots E_{d-i}) \tag{2}$$

where E_i is the event "extract an inlier at the i-th drawing". In the case of uniform sampling $P(E_i|E_1, E_2\dots E_{i-1}) = \frac{S-i+1}{N-i+1}$. In our case, the first point is sampled with uniform probability, hence $P(E_1) = S/N$, while the others are sampled with the probability function (1), therefore, after expanding the normalization constant Z, the conditional probability can be approximated as

$$P(E_i|E_1, E_2\dots E_{i-1}) = \frac{(S-i+1)\exp -\frac{\alpha^2}{\sigma^2}}{(N-S-i+1)\exp -\frac{\omega^2}{\sigma^2} + (S-i+1)\exp -\frac{\alpha^2}{\sigma^2}} \quad i = 2\dots d \tag{3}$$

where α is the average inlier-inlier distance, and ω is the average inlier-outlier distance. If $S \gg d$ then

$$p \simeq \delta \left(\frac{\delta \exp -\frac{\alpha^2}{\sigma^2}}{(1-\delta)\exp -\frac{\omega^2}{\sigma^2} + \delta \exp -\frac{\alpha^2}{\sigma^2}} \right)^{d-1}. \tag{4}$$

where $\delta = S/N$ is the inlier fraction for a given model. Therefore, assuming that ω is larger than α, the sampling strategy increases the probability of extracting an outlier-free MSS, as the intuition would also suggests.

Finally, the probability of drawing at least K outlier-free MSS out of M, for a given model, is given by [7]:

$$\rho = 1 - \sum_{k=0}^{K-1} \binom{M}{k} p^k (1-p)^{M-k}. \tag{5}$$

This equation is used to compute the required number of samples M for a given confidence ρ and a given K. The value of δ in (4) must be set to the smallest inliers fraction among all the models. Values of ρ vs M are shown in Fig. 2.

2.2 J-Linkage Clustering

Models are extracted by agglomerative clustering of data points in the conceptual space, where each point is represented by (the characteristic function of) its preference set.

The general agglomerative clustering algorithm proceeds in a bottom-up manner: Starting from all singletons, each sweep of the algorithm merges the two

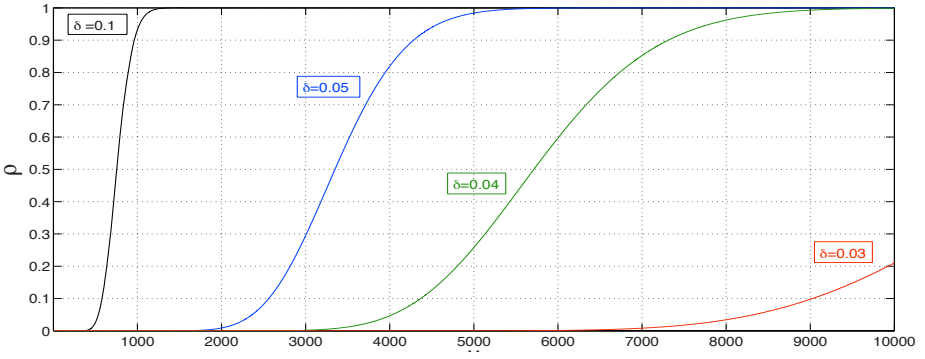


Fig. 2. Plot of ρ vs M for different values of δ with $K = 25, \alpha^2 = 0.5\sigma^2\beta^2 = 3.0\sigma^2$

clusters with the smallest distance. The way the distance between clusters is computed produces different flavours of the algorithm, namely the simple linkage, complete linkage and average linkage [10].

We propose a variation that fits very well to our problem, called *J-linkage* (see Algorithm 1). First the preference set of a cluster is computed as the *intersection* of the preference sets of its points. Then the distance between two elements (point or cluster) is computed as the *Jaccard distance* between the respective preference sets.

Definition 1 (Jaccard distance). *Given two sets A and B, the Jaccard distance is*

$$d_{\mathrm{J}}(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}.$$

The Jaccard distance measures the degree of overlap of the two sets and ranges from 0 (identical sets) to 1 (disjoint sets).

The cut-off value is set to 1, which means that the algorithm will only link together elements whose preference sets overlap. Please note that the cut-off distance is not data dependent, but defines a qualitative behaviour of the J-linkage algorithm. Indeed, as a result, clusters of points have the following properties:

- for each cluster there exist at least one models that is in the PS of all the points (i.e., a model that fits all the points of the cluster)
- one model cannot be in the PS of *all* the points of two distinct clusters (otherwise they would have been linked).

Each cluster of points defines (at least) one model. If more models fit all the points of a cluster they must be very similar. The final model for each cluster of points is estimated by least squares fitting.

Outliers emerge as small clusters. Depending on the application, one may set different rejection thresholds. If the percentage of outliers is known or can be estimated (as it is assumed in RANSAC), one may reject all the smallest clusters up to the number of outliers.

Algorithm 1. J-LINKAGE

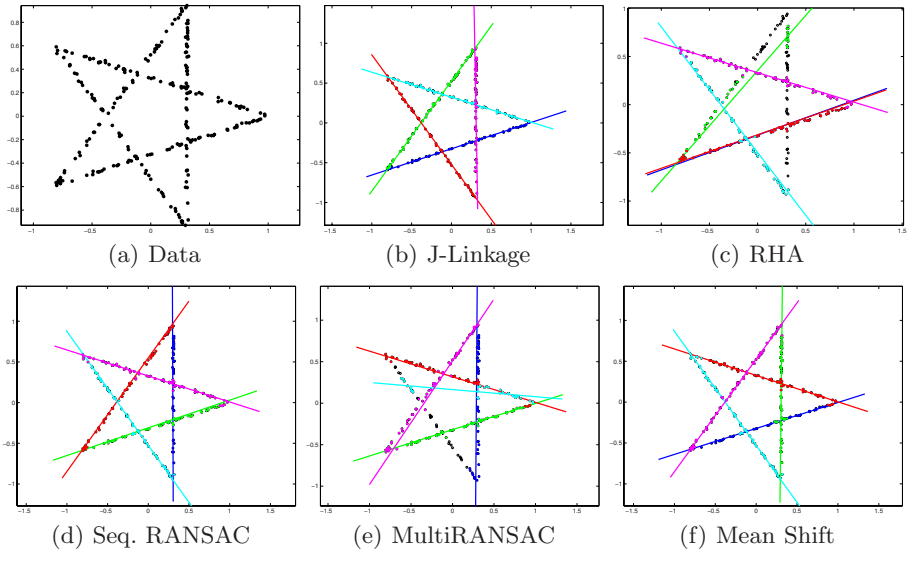
Input: the set of data points, each point represented by its preference set (PS)
Output: clusters of points belonging to the same model

1. Put each point in its own cluster.
2. Define the PS of a cluster as the *intersection* of the PSs of its points.
3. Among all current clusters, pick the two clusters with the smallest Jaccard distance between the respective PSs.
4. Replace these two clusters with the union of the two original ones.
5. Repeat from step 3 while the smallest Jaccard distance is lower than 1.

3 Experiments

We performed comparative experiments with sequential RANSAC, multi RANSAC, residual histogram analysis [7] (henceforth RHA) and mean-shift (MS). In all the experiments each model consists of 50 inliers, corrupted by variable Gaussian noise and variable outliers percentage. The data sets consist of segments in several configuration: star (*star5* and *star11*), circles (*circle5*), and horizontal (*stair4*). The latter was used also in [2].

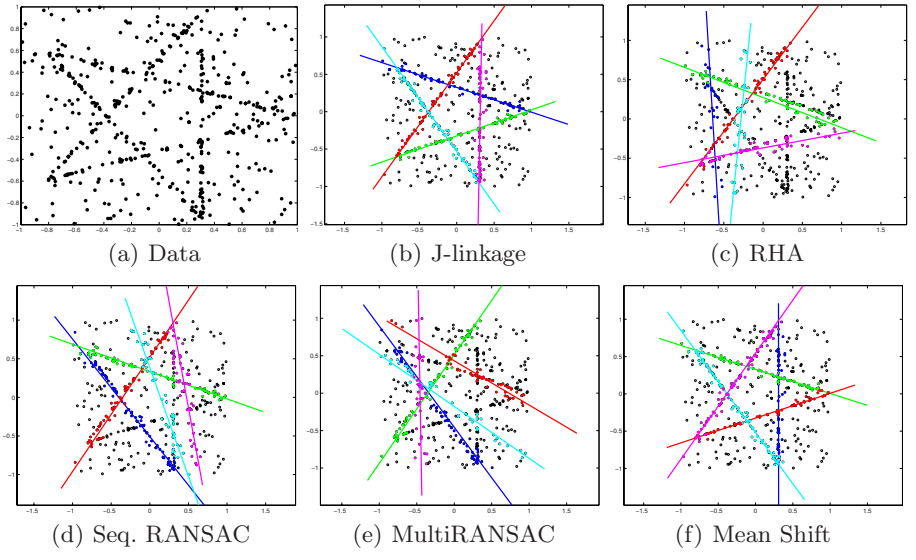
All the methods being compared are based on random sampling, hence we used the same sampling strategy (Eq. 1) and number of samples (5000) in all the experiments. The scale parameter σ in the sampling strategy is 0.2 in all the experiments but *stair4*, where it has been set to 0.05. The inlier threshold ε – variable with the noise level – was the same for sequential RANSAC, multi-RANSAC and our method. The parameters needed by MS (bandwidth) and by



(a) Data (b) J-Linkage (c) RHA

(d) Seq. RANSAC (e) MultiRANSAC (f) Mean Shift

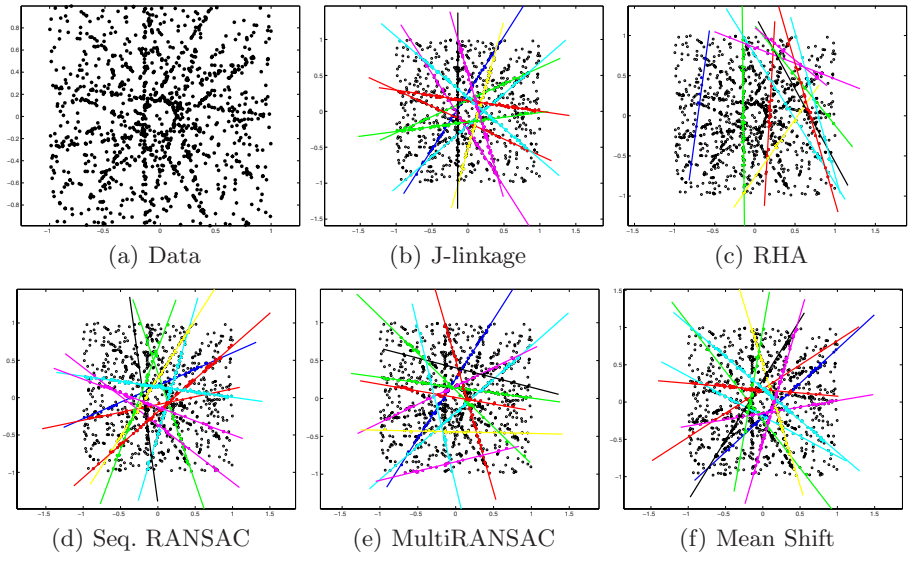
Fig. 3. *star5* set perturbed with Gaussian noise ($\sigma_n = 0.0075$) and no outliers



(a) Data (b) J-linkage (c) RHA

(d) Seq. RANSAC (e) MultiRANSAC (f) Mean Shift

Fig. 4. *star5* set perturbed with Gaussian noise ($\sigma_n = 0.0075$) and 50% outliers



(a) Data (b) J-linkage (c) RHA

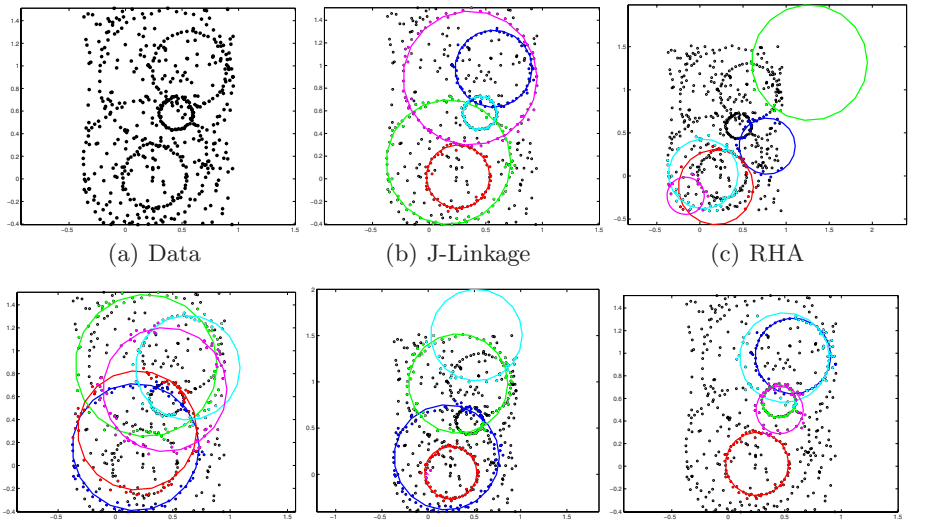
(d) Seq. RANSAC (e) MultiRANSAC (f) Mean Shift

Fig. 5. *star11* set perturbed with Gaussian noise ($\sigma_n = 0.0075$) and 50% outliers

RHA have been tuned manually to optimize performances. The best outcome out of several trials have been recorded. As multiRANSAC requires prior specification of the number n of models, for the sake of fairness we used the same information also with the other algorithms: only the best or strongest n models among the ones produced by the algorithm were considered. For example, with

J-linkage we retained the n largest clusters, and the same for MS. In RHA we sought the n strongest modes.

The results on synthetic data are reported in Figures 3, 4, 5, 6, 7. In summary, we see that MS on the *star5* data set always produces the correct result, whereas all the other methods break when the outlier percentage grows to 50%. If the number of models increases (*star11* data set), however, only J-linkage produces the correct result, at the same level of noise and outliers. Also on the *circle5* data set, with random noise and 50% outliers, J-linkage is the only one that works correctly. On the *stair4* data set with and 60% outliers both multiRANSAC and J-linkage yield the correct result.



(a) Data (b) J-Linkage (c) RHA

(d) Seq. RANSAC (e) MultiRANSAC (f) Mean Shift

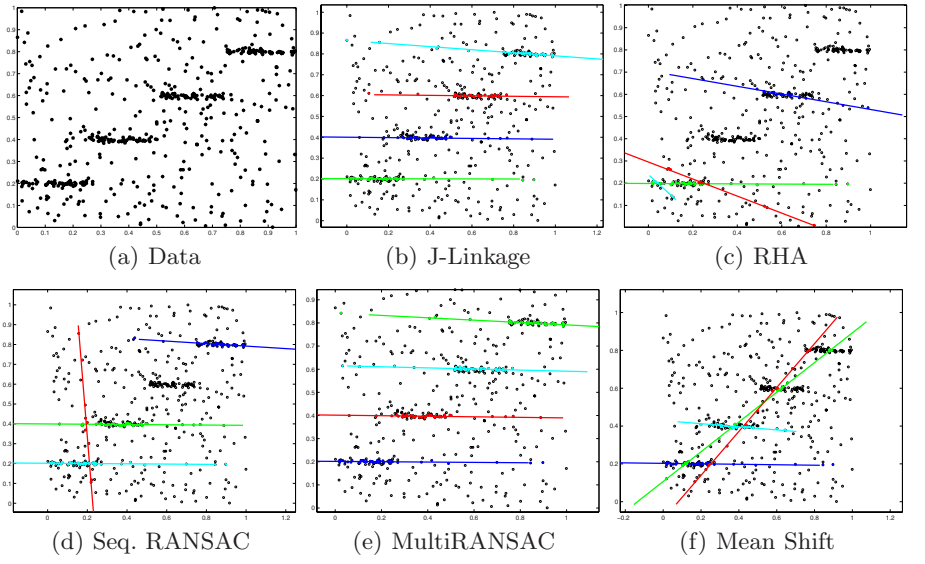
Fig. 6. *circle5* set perturbed with Gaussian noise ($\sigma_n = 0.0075$) and 50% outliers

We noted that multiRANSAC systematically fails when the models intersect each other (the problem is more evident when the models are cyclically intersecting). The reason is that the greedy approach adopted by multiRANSAC tries to maximize the number of total inliers, implicitly assuming non intersecting models.

As an example of a real case, Fig. 8 shows the results of fitting planes to a cloud of 3D points. They were produced by a Structure and Motion pipeline [11] fed with a set of images of the church of Pozzoveggiani (Italy). Despite the fact that gross outliers are absent, pseudo-outliers and the uneven distribution of points among the models (ranging from 9 to 1692) challenges any model fitting algorithm. Indeed, our method is the only one that produces the correct result. To appraise further the result of J-linkage, Fig. 8 show two images of Pozzoveggiani church with points marked according to the plane they belong to.

The MATLAB code is available for download from the web[2].

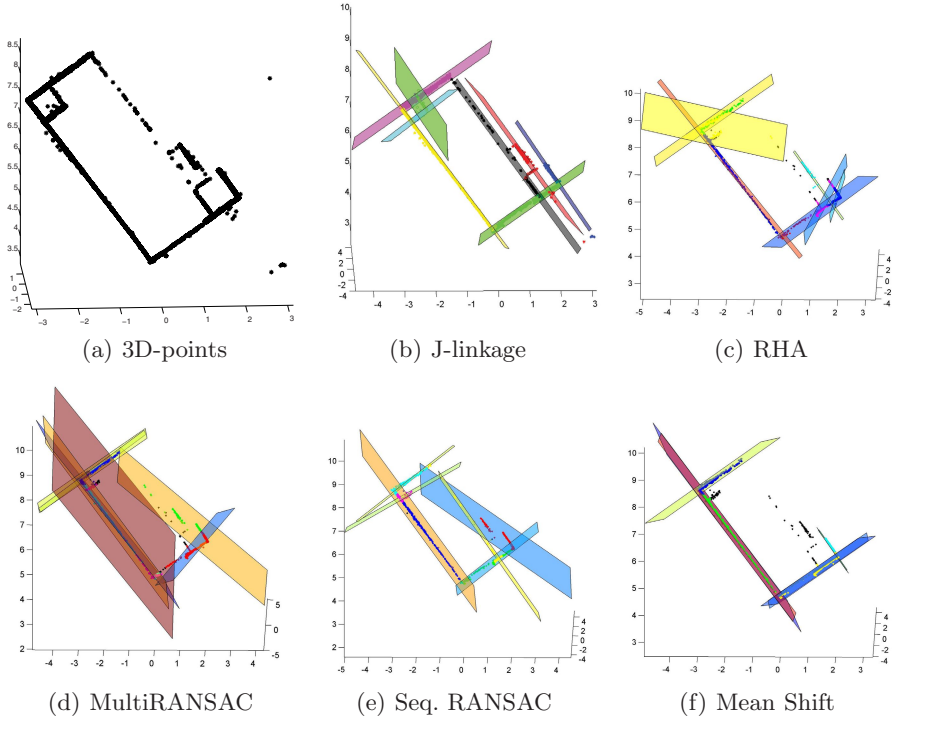
[2] http://profs.sci.univr.it/~fusiello/demo/jlk/



(a) Data (b) J-Linkage (c) RHA

(d) Seq. RANSAC (e) MultiRANSAC (f) Mean Shift

Fig. 7. *stair4* set perturbed with Gaussian noise ($\sigma_n = 0.0075$) and 60% outliers



(a) 3D-points (b) J-linkage (c) RHA

(d) MultiRANSAC (e) Seq. RANSAC (f) Mean Shift

Fig. 8. "Pozzoveggiani" dataset. 3D planes extracted by J-linkage and other algorithms viewed from the zenith.

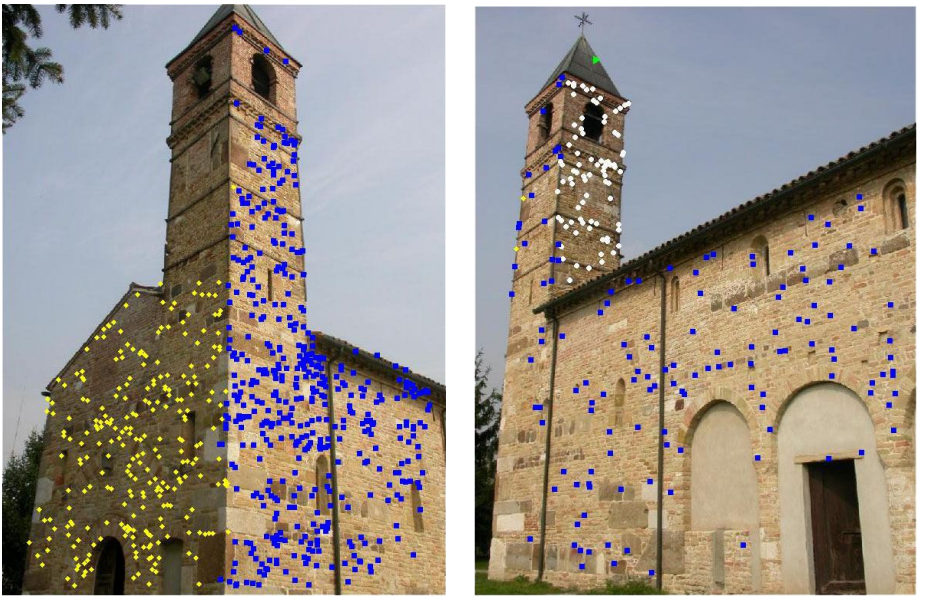


Fig. 9. Two images of the "Pozzoveggiani" dataset. Points belonging to different planes according to J-linkage are plotted with different markers.

4 Discussion

The experiments showed that J-linkage compares favorably with other state-of-the-art methods. Our insight about the reason for this is that it derives the best features from the competing algorithms. Like RANSAC it is robust because only the points below the outlier threshold matters. Compared to RHA, it consider only the the first bin of the residuals histogram, while RHA takes the whole the histogram into account, which is corrupted by outliers. Like RHA, however, our conceptual representation casts the problem from the perspective of the data point, which is a strength of RHA.

The discovery of multiple models is devolved to clustering, thereby gaining a global view of the problem, whereas sequential RANSAC and multiRANSAC are forced to make local, greedy choices. Clustering, however, is not an intrinsically robust technique, in general. J-linkage, instead, is inherently robust, because it is based on the intersection of preference sets, hence it favours the growing of clusters made up of inliers only. On the contrary, modes of the residual histogram or of the parameters distribution are difficult to locate, especially when the number of gross outliers and pseudo-outliers grows, and the fraction of outlier-free sets generated by the random sampling decreases accordingly.

5 Conclusions

We described a novel method for fitting multiple instances of a model to data corrupted by noise and outliers. Each point is represented with the characteristic function of its preference set and multiple models are revealed as clusters in this conceptual space. A new agglomerative clustering algorithm, called J-linkage, have been specifically devised. The method does not require prior specification of the number of models, nor it necessitate manual parameters tuning. The only free parameter is the consensus threshold, as in RANSAC. Our method demonstrated its effectiveness in comparison with state-of-the-art competing algorithms.

Acknowledgements. This research was supported by the Italian PRIN 2006 project 3-SHIRT.

References

1. Stewart, C.V.: Bias in robust estimation caused by discontinuities and multiple structures. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(8), 818–833 (1997)
2. Zuliani, M., Kenney, C.S., Manjunath, B.S.: The multiRANSAC algorithm and its application to detect planar homographies. In: Proceedings of the IEEE International Conference on Image Processing, Genova, IT, September 11-14 (2005)
3. Xu, L., Oja, E., Kultanen, P.: A new curve detection method: randomized Hough transform (RHT). Pattern Recognition Letters 11(5), 331–338 (1990)
4. Subbarao, R., Meer, P.: Nonlinear mean shift for clustering over analytic manifolds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, York, USA, pp. 1168–1175 (2006)
5. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5), 603–619 (2002)
6. Vidal, R., Ma, Y., Sastry, S.: Generalized principal component analysis (gpca). IEEE Transactions on Pattern Analysis and Machine Intelligence 27(12), 1945–1959 (2005)
7. Zhang, W., Kosecká, J.: Nonparametric estimation of multiple structures with outliers. In: Vidal, R., Heyden, A., Ma, Y. (eds.) WDV 2006. LNCS, vol. 4358, pp. 60–74. Springer, Heidelberg (2006)
8. Duin, R., Pekalska, E., Paclik, P., Tax, D.: The dissimilarity representation, a basis for domain based pattern recognition? In: Goldfarb, L. (ed.) Pattern representation and the future of pattern recognition, ICPR 2004 Workshop Proceedings, Cambridge, UK, pp. 43–56 (2004)
9. Kanazawa, Y., Kawakami, H.: Detection of planar regions with uncalibrated stereo using distributions of feature points. In: British Machine Vision Conference, pp. 247–256 (2004)
10. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis, pp. 98–105. John Wiley and Sons, Chichester (1973)
11. Farenzena, M., Fusiello, A., Gherardi, R.: Efficient Visualization of Architectural Models from a Structure and Motion Pipeline. In: Eurographics 2008 - Short Papers, Crete, Greece, Eurographics Association, pp. 91–94 (2008)

Human Activity Recognition with Metric Learning

Du Tran and Alexander Sorokin

University of Illinois at Urbana-Champaign
Urbana, IL, 61801, USA
{dutran2,sorokin2}@uiuc.edu

Abstract. This paper proposes a metric learning based approach for human activity recognition with two main objectives: (1) reject unfamiliar activities and (2) learn with few examples. We show that our approach outperforms all state-of-the-art methods on numerous standard datasets for traditional action classification problem. Furthermore, we demonstrate that our method not only can accurately label activities but also can reject unseen activities and can learn from few examples with high accuracy. We finally show that our approach works well on noisy YouTube videos.

1 Introduction

Human activity recognition is a core unsolved computer vision problem. There are several reasons the problem is difficult. First, the collection of possible activities appears to be very large, and no straightforward vocabulary is known. Second, activities appear to compose both across time and across the body, generating tremendous complexity. Third, the configuration of the body is hard to transduce, and there is little evidence about what needs to be measured to obtain a good description of activity.

Activity can be represented with a range of features. At low spatial resolution when limbs cannot be resolved, flow fields are discriminative for a range of motions [1]. At higher spatial resolutions one can recover body configuration and reason about it [2,3]. There is strong evidence that 3D configuration can be inferred from 2D images (e.g. [4,5,6]; see also discussion in [7]), which suggests building appearance features for body configuration. Such appearance features include: braids [8]; characteristic spatio-temporal volumes [9]; motion energy images [10]; motion history images [10]; spatio-temporal interest points [11,12,13]; nonlinear dimensionality reduced stacks of silhouettes [14]; an extended radon transform [15]; and silhouette histogram of oriented rectangle features [16]. Generally, such features encode (a) what the body looks like and (b) some context of motion. We follow this general pattern with some innovations (Section 2).

An activity recognition process should most likely have the following properties: **Robustness:** features should be relatively straightforward to obtain from sequences with reasonable accuracy, and should demonstrate good noise behaviour. **Discriminative:** at least for the primitives, one would like discriminative

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 548–561, 2008.

rather than generative models, so that methods can focus on what is important about the relations between body configuration and activity and not model irrelevant body behaviour. **Rejection:** activity recognition is going to be working with a set of classes that is not exhaustive for the forseeable future; this means that when a system encounters an unknown activity, it should be labelled unknown.

The whole set of requirements is very demanding. However, there is some evidence that activity data may have the special properties needed to meet them. First, labelling motion capture data with activity labels is straightforward and accurate [17]. Second, clustering multiple-frame runs of motion capture data is quite straightforward, despite the high dimensions involved, and methods using such clusters do not fail (e.g. [18]). Third, motion capture data compresses extremely well [19]. All this suggests that, in an appropriate feature space, motion data is quite easy to classify, because different activities tend to look quite strongly different. Following that intuition, we argue that a metric learning algorithm (e.g. [20,21,22,23]) can learn an affine transformation to a good discriminative feature space even using simple and straightforward-to-compute input features.

1.1 Contributions of the Paper

This paper has the following contributions:

1. Proposes a metric learning based approach for human activity recognition with the abilities to reject unseen activities and to learn with few training examples (Sections 3.4, 5.2).
2. Provides a large body of experimental evidence showing that quite simple appearance features (Section 2) work better than more complex ones (Section 5.1).
3. Demonstrates that our approach achieves strong results on a realistic dataset despite the noise (Section 6).

2 Motion Context Descriptor

Local descriptor. Our frame descriptor is a histogram of the silhouette and of the optic flow inside the normalized bounding box. We scale the bigger side of the bounding box to a fixed size $M (M = 120)$ preserving the aspect ratio. The scaled box is then placed at the center bottom of an $M \times M$ square box padded with zeros. We use this transformation to resample the values of the flow vectors and of the silhouette.

The optic flow measurements are split into horizontal and vertical channels. To reduce the effect of noise, each channel is smoothed using median filter. This gives us two real-valued channels F_x and F_y. The silhouette gives us the third (binary) channel S. Each of the 3 channels is histogrammed using the same technique: The normalized bounding box is divided into 2×2 sub-windows. Each sub-window is then divided into 18 pie slices covering 20 degrees each. The

center of the pie is in the center of the sub-window and the slices do not overlap. The values of each channel are integrated over the domain of every slice. The result is a $72(2\times2\times18)$-dimensional histogram. By concatenating the histograms of all 3 channels we get a 216-dimensional frame descriptor.

In our experiments, we also experimented with 3×3 and 4×4 sub-windows. 3×3 is not different from 2×2, but 4×4 decreases the performance by 5-7%. The radial histograms are meaningless when the sub-windows are getting too small.

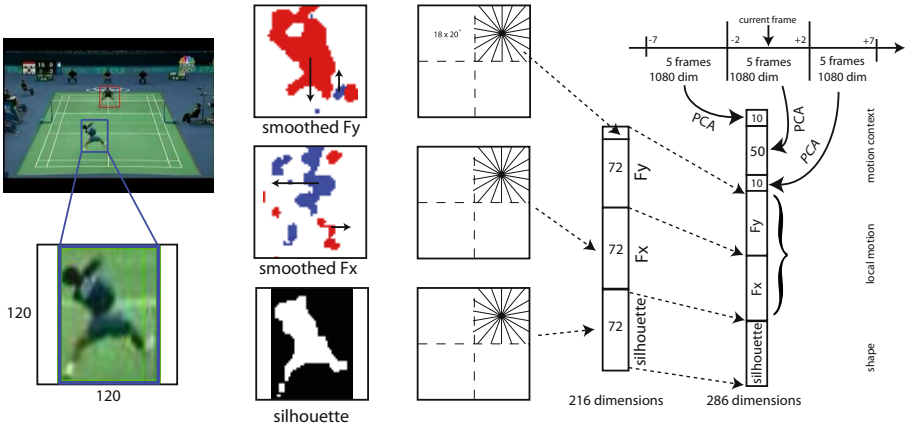


Fig. 1. Feature Extraction: The three information channels are: vertical flow, horizontal flow, silhouette. In each channel, the measurements are resampled to fit into normalized (120×120) box while maintaining aspect ratio. The normalized bounding box is divided into 2x2 grid. Each grid cell is divided into 18-bin radial histogram (20^o per bin). Each of the 3 channels is separately integrated over the domain of each bin. The histograms are concatenated into 216 dimensional frame descriptor. 5-frame blocks are projected via PCA to form medium scale motion summaries. The first 50 dimensions are kept for the immediate neighborhood and the first 10 dimensions are kept for each of the two adjacent neighborhoods. The total 70-dimensional motion summary is added to the frame descriptor to form the motion context.

Motion context. We use 15 frames around the current frame and split them into 3 blocks of 5 frames: past, current and future. We chose a 5-frame window because a triple of them makes a 1-second-long sequence (at 15 fps). The frame descriptors of each block are stacked together into a 1080 dimensional vector. This block descriptor is then projected onto the first N principal components using PCA. We keep the first 50, 10 and 10 dimensions for the current, past and future blocks respectively. We picked 50, 10 and 10 following the intuition that local motion should be represented in better detail than more distant ones. The resulting 70-dimensional context descriptor is appended to the current frame descriptor to form the final 286-dimensional motion context descriptor.

We design our features to capture local appearance and local motions of the person. Our Motion Context descriptor borrows the idea of radial bins from the

Shape Context [24] and of the noisy optic flow measurements from the "30-pixel man" [1]. We append a summary of the motion around the frame to represent medium-scale motion phenomena. We assume that the bounding box of the actor together with the silhouette mask is provided. In this work we use background subtraction to obtain the silhouette and the bounding box. These are often noisy, however our feature representation seems to be tolerant to some level of noise. Our experiments with badminton sequences show that when the noise is too extreme, it starts to affect the accuracy of activity recognition. We compute optic flow using Lucas-Kanade algorithm [25].

3 Action Classification Models

3.1 Naïve Bayes

Naïve Bayes requires the probability $P(frame|l)$ of the frame given the label l. To compute this probability we apply vector quantization via K-Means. After vector quantization the frame is represented by a word w_i and the probability is estimated by counting with Laplace smoothing: $P(w_i|l) = \frac{c(w_i,l)+1}{c(w,l)+K}$ where $c(w_i, l)$ is the numbers of times the word w_i occurred with the label l and $c(w, l)$ is the total number of words with the label l. Assuming uniform prior $P(l)$, ignoring $P(seq)$ and using Bayes rule we get the following prediction rule:

$$l = \operatorname*{argmax}_l P(l|seq) = \operatorname*{argmax}_l \sum_{i=1}^{m} \log P(w_i|l) \tag{1}$$

3.2 1-Nearest Neighbor

1NN classifier assigns a label to every query frame by finding the closest neighbor among training frames and propagating the label from the neighbor to the query frame. Every frame of the query sequence then votes for the label of the sequence. The label of the sequence is determined by the majority. Note that the voting provides us with smoothing and robustness to noise and thus we do not need to use more than one nearest neighbor.

3.3 1-Nearest Neighbor with Rejection

Nearest Neighbors with Rejection work by fixing a radius R and ignoring points further than R. If no neighbor is found within R, the query frame is thus unseen and receives the label "unobserved". The sequence is then classified by the majority vote including the "unobserved" label. We also consider the classifier that does rejection after metric learning. We manually choose the rejection radius to achieve equal accuracy on the discriminative and rejection tasks. The rejection radius can be chosen by cross validation to achieve desired trade-off between the discriminative and rejection tasks.

3.4 1-Nearest Neighbor with Metric Learning

Nearest neighbors crucially depend on the metric of the embedding space. Among metric learning algorithms ([20,21,22,23]), Large Margin Nearest Neighbors (LMNN) [22] are especially tailored to k-NN classifiers. We briefly state LMNN below.

LMNN learns a Mahalanobis distance D:

$$D(x_i, x_j) = (x_i - x_j)^T M(x_i - x_j) = \|L(x_i - x_j)\|^2 \tag{2}$$

LMNN tries to learn a matrix $M = L^T L$ that maximizes the distances between examples with different labels and minimizes the distances between nearby examples with the same label.

Minimize:

$$\sum_{ij} \eta_{ij}(x_i - x_j)^T M(x_i - x_j) + c \sum_{ijl} \eta_{ij}(1 - y_{il})\xi_{ijl}$$

Subject to:

$$\begin{array}{ll} (i) & (x_i - x_l)^T M(x_i - x_l) - (x_i - x_j)^T M(x_i - x_j) \geq 1 - \xi_{ijl} \\ (ii) & \xi_{ijl} \geq 0 \\ (iii) & M \succeq 0 \end{array} \tag{3}$$

where y_{ij} is a binary value indicating whether points x_i and x_j are in the same class and η_{ij} is a binary value indicating whether x_j is a selected nearby neighbor of x_i with the same class, ξ_{ijl} are slack variables. In the objective function, the first term minimizes the distances between all training examples and their selected neighbors. The second term maximizes the margin (relaxed by slack variables) between same-label distances (x_i to x_j) and different-label distances (x_i to x_l) of all training examples. We used the source code kindly provided by the authors of [22].

LMNN learns a global transformation matrix, but its objective is designed to capture the local manifold structure by selecting k nearby neighbors. Normally k is small and in our experiments, we use $k = 3$.

Data subsampling. We note that it is important to subsample training data before applying metric learning. Applying metric learning without subsampling training data will not help in discriminative task and even decreases the performance by 6-8%. This phenomenon is easy to understand. Without subsampling the training examples, the k selected neighbors of every frame are always the neighboring frames from the same sequence. Therefore minimizing the distances between examples with the same label is not helpful. In our experiment, we subsample training examples by the ratio 1:4, choosing 1 from every 4 consecutive frames.

LMNN significantly improves recognition accuracy when it operates in the complete feature space. However it is computationally expensive. We studied the improvements produced by LMNN if the feature space is restricted to be low-dimensional. There are two immediately obvious ways to reduce dimensionality: PCA and random projections (we use [26]). We used a range of dimensions from 5 to 70 with a step of 5. The results are discussed in Section 5.2.

4 Experimental Setup

4.1 Description of the Datasets

For our experiments we used **5 datasets**: 3 datasets presented in the literature and 2 new datasets. The **Weizman dataset** [9] contains 81 isolated sequences of 9 actors performing 9 activities. We use an augmented and more difficult version with 93 isolated sequences of 9 actors and 10 activities with 3 extra sequences. The **UMD dataset** [27] contains 100 sequences of 10 activities performed 10 times each by only one actor. The **IXMAS dataset** [28] contains 36 sequences in which 12 actors perform 13 actions. Each sequence is captured in 5 different views. **Our dataset 1** consists of 532 high resolution sequences of 14 activities performed by 8 actors. **Our dataset 2** consists of 3 badminton sequences downloaded from Youtube. The sequences are 1 single and 2 double matches at the Badminton World Cup 2006.

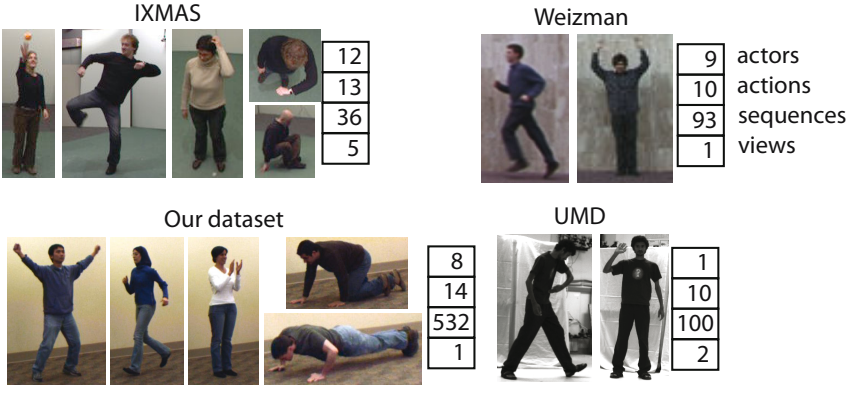


Fig. 2. The variations in the activity dataset design: **Weizman**: multiple actors, single view and only one instance of activity per actor, low resolution (80px). **Our**: multiple actors, multiple actions, extensive repetition, high resolution (400px), single view. **IXMAS**: Multiple actors, multiple synchronized views, very short sequences, medium-low resolution (100,130,150,170,200px). **UMD**: single actor, multiple repetitions, high resolution (300px).

4.2 Evaluation Protocols

We evaluate the accuracy of the activity label prediction for a query sequence. Every sequence in a dataset is used as a query sequence. We define 7 evaluation protocols by specifying the composition of the training set w.r.t. the query sequence. Leave One Actor Out (**L1AO**) excludes all sequences of the same actor from the training set. Leave One Actor-Action (**L1AAO**) excludes all sequences matching both action and actor with the query sequence. Leave One View Out (**L1VO**) excludes all sequences of the same view from the training set. This

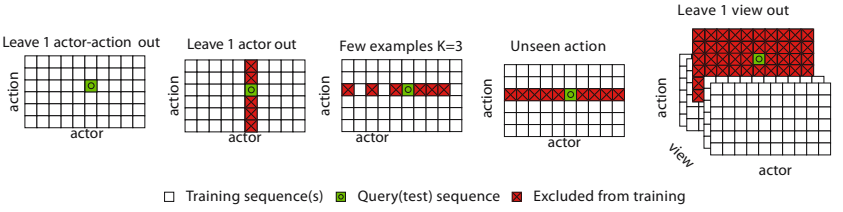


Fig. 3. Evaluation protocols: **Leave 1 Actor Out** removes all sequences of the same actor from the training set and measures prediction accuracy. **Leave 1 Actor-Action Out** removes all examples of the query activity performed by the query actor from the training set and measures prediction accuracy. This is more difficult task than L1AO. **Leave 1 View Out** measures prediction accuracy across views . **Unseen Action** removes all examples of the same action from the training set and measures rejection accuracy. **Few Examples-K** measures average prediction accuracy if only K examples of the query action are present in the training set. Examples from the same actor are excluded.

protocol is only applicable for datasets with more than one view(UMD and IXMAS). Leave One Sequence Out (**L1SO**) removes **only** the query sequence from the training set. If an actor performs every action once this protocol is equivalent to L1AAO, otherwise it appears to be easy. This implies that vision-based interactive video games are easy to build. We add two more protocols varying the number of labeled training sequences. Unseen action (**UAn**) protocol excludes from the training set all sequences that have the same action as the query action. All other actions are included. In this protocol the correct prediction for the sequence is not the sequence label, but a special label "reject". Note that a classifier always predicting "reject" will get 100% accuracy by UAn but 0% in L1AO and L1AAO. On the contrary, a traditional classifier without "reject" will get 0% accuracy in UAn.

Few examples (**FE-K**) protocol allows K examples of the action of the query sequence to be present in the training set. The actors of the query sequences are required to be different from those of training examples. We randomly select K examples and average over 10 runs. We report the accuracy at K=1,2,4,8. Figure 3 shows the example training set masks for the evaluation protocols.

5 Experimental Results

5.1 Simple Feature Outperforms Complex Ones

We demonstrate that our approach achieves state of the art discriminative performance. Table 2 compares our performance with published results. We show that on a large number of standard datasets with closed world assumption we easily achieve state-of-the-art perfect accuracy. Note that there are two versions of Weizman dataset, the original one contains 9 actions while the augmented version has 10. Our model achieves perfect accuracy on both Weizman datasets. For UMD dataset, we find that, it is easy to achieve 100% accuracy with train

Table 1. Experimental Results show that conventional discriminative problems **L1AAO,L1AO,L1SO** are easy to solve. Performance is in the high 90's (consistent with the literature). Learning with few examples **FE-K** is significantly more difficult. Conventional discriminative accuracy is not a good metric to evaluate activity recognition, where one needs to refuse to classify novel activities. Requiring rejection is expensive; the objective **UNa** decreases discriminative performance. In the table bold numbers show the best performance among rejection-capable methods. **N/A** denotes the protocol being inapplicable or not available due to computational limitations.

Dataset	Algorithm	Chance	Protocols								
			Discriminative task				Reject	Few examples			
			L1SO	L1AAO	L1AO	L1VO	UNa	FE-1	FE-2	FE-4	FE-8
Weizman	NB(k=300)	10.00	91.40	93.50	95.70	N/A	0.00	N/A	N/A	N/A	N/A
	1NN	10.00	95.70	95.70	96.77	N/A	0.00	53.00	73.00	89.00	96.00
	1NN-M	10.00	100.00	100.00	100.00	N/A	0.00	72.31	81.77	92.97	100.00
	1NN-R	9.09	83.87	84.95	84.95	N/A	84.95	17.96	42.04	68.92	84.95
	1NN-MR	9.09	**89.66**	**89.66**	**89.66**	N/A	**90.78**	N/A	N/A	N/A	N/A
Our	NB(k=600)	7.14	98.70	98.70	98.70	N/A	0.00	N/A	N/A	N/A	N/A
	1NN	7.14	98.87	97.74	98.12	N/A	0.00	58.70	76.20	90.10	95.00
	1NN-M	7.14	**99.06**	**97.74**	**98.31**	N/A	0.00	88.80	94.84	95.63	98.86
	1NN-R	6.67	95.86	81.40	82.10	N/A	81.20	27.40	37.90	51.00	65.00
	1NN-MR	6.67	**98.68**	**91.73**	**91.92**	N/A	**91.11**	N/A	N/A	N/A	N/A
IXMAS	NB(k=600)	7.69	80.00	78.00	79.90	N/A	0.00	N/A			
	1NN	7.69	81.00	75.80	80.22	N/A	0.00				
	1NN-R	7.14	**65.41**	**57.44**	**57.82**	N/A	**57.48**				
UMD	NB(k=300)	10.00	100.00	N/A	N/A	97.50	0.00	N/A			
	1NN	10.00	100.00	N/A	N/A	97.00	0.00				
	1NN-R	9.09	**100.00**	N/A	N/A	**88.00**	**88.00**				

Table 2. Accuracy Comparison shows that our method achieves state of the art performance on large number of datasets. *-full 3D model (i.e. multiple camera views) is used for recognition.

Dataset	Weizman9						Weizman10			UMD			IXMAS		
Method	[29]	[30]	[31]	[9]	[16]	**Our**	[32]	[14]	**Our**	[27]	[14]	**Our**	[33]	[28]*	**Our**
Accuracy	72.8	92.6	98.8	99.67	**100**	**100**	82.6	97.78	**100**	**100**	**100**	**100**	80.06	**93.33**	**81**

and test on the same actor, playing the same action in the same view. In this case even L1VO achieved 97.5% accuracy on this dataset. On IXMAS dataset, [28] report higher (93.33%) accuracy, however they use full 3D model.

5.2 Metric Learning Improves Action Classification

We demonstrate that metric learning significantly improves human activity recognition performance in: (1) discriminative task, (2) rejection task, and (3) few examples. On traditional action recognition problem, 1NN-M achives almost perfect accuracy and outperforms all state-of-the-art methods. For rejection task, 1NN-MR improves the accuracy about 5% on Weizman dataset and 10% on our dataset comparing to 1NN-R. For learning with few examples, 1NN-M significantly improves the accuracy. Specifically, for 1-example, 1NN-M improves

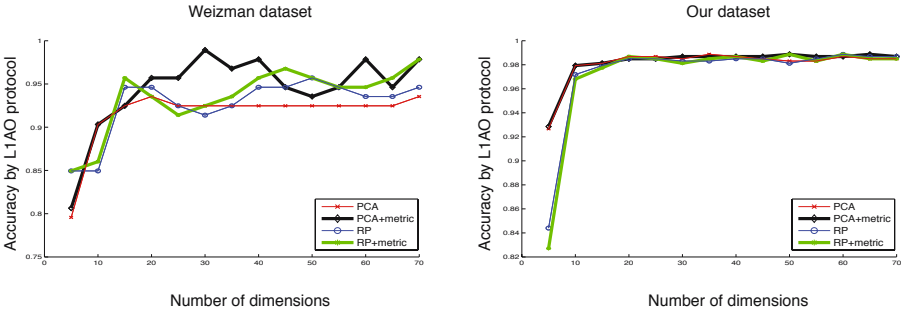


Fig. 4. LMNN with Dimension Reduction: On Weizman dataset, LMNN clearly improves PCA ($2.8 \pm 2.0\%$) and almost improves random projections ($0.8 \pm 1.2\%$). On our dataset, LMNN improvements are not present with few dimensions on the closed world classification task ($0.1 \pm 0.2\%$ from PCA and $0.1 \pm 0.5\%$ from random projection); The improvement is 2% in high dimensions and 3%-10% in rejection task.

about 20% accuracy on Weizman dataset and 30 % accuracy on our dataset. We show that our approach achieves about 72.31% accuracy on Weizman dataset and 88.80% on our dataset for action classification with only one training example. In low dimensions there is not much benefit from LMNN (Fig 4). The only clear improvement appears on Weizman dataset with PCA. In other cases of low dimensionality produce very little improvement if any.

6 Video Labeling with Rejection

How would we spot activities in practice? We would take a video, label some of the example activities and propagate the labels to unseen video. We follow this scenario and apply our algorithm to Youtube videos. We work with 3 badminton match sequences: 1 single and 2 double matches of the Badminton World Cup 2006.

Fig. 5. Our Dataset 2: Example frames from badminton sequences collected from Youtube. The videos are low resolution (60-80px) with heavy compression artifacts. The videos were chosen such that background subtraction produced reasonable results.

Table 3. Label sets for badminton action recognition

Problem	Label set
1. Type of motion	*run, walk, hop, jump, unknown*
2. Type of shot	*forehand, backhand, smash, unknown*
3. Shot detection	*shot, non-shot*

For a badminton match we define 3 human activity recognition problems shown in table 3. Problem 1 is to classify the type of motion of each player. Problem 2 is to classify the shot type. Problem 3 is to predict the moment of the shot. The players closer to the camera are very different from the players in the back. We therefore define two different "views" and evaluate the labeling performance for each view separately as well as for both views combined. One of the sequences was manually labeled for training and quantitative evaluation. The first half of the sequence is used for training, while the second half is used for testing. For problems 1 and 2 we measure prediction accuracy. For problem 3 we measure the distance from the predicted shot instant to the labeled one.

Table 4. Quantitative evaluation of video labeling show the prediction accuracy of **1NN**, **1NN-R**, **1NN-M**, and **1NN-MR** for the video labeling task. One Youtube sequence was manually annotated. The first half was used for training, the second half for evaluation. View 1 shows significantly better results due to higher resolution on the person giving more stable segmentation and less noisy flow computation. **1NN** works well in the closed world. However it performs poorly when it is applied to the open world. The underlined performance (in red) is below chance. **1NN-M** improves 1-6% from **1NN**. **1NN-MR** improves 0.5-4% from **1NN-R** in "view 2" but the other views.

Problem	Algo	View 1	View 2	2 Views	Chance	Assumption
1. Motion	1NN	75.81	63.66	71.30	25.00	close
2. Shot	1NN	88.84	81.50	74.55	33.33	close
1. Motion	1NN-M	**76.46**	**69.25**	**71.89**	25.00	close
2. Shot	1NN-M	**89.52**	**86.23**	**78.82**	33.33	close
1. Motion	1NN	42.72	24.93	34.04	20.00	open
2. Shot	1NN	26.49	23.75	21.98	25.00	open
1. Motion	1NNR	**57.73**	47.95	**53.37**	20.00	open
2. Shot	1NNR	**63.45**	52.29	52.15	25.00	open
1. Motion	1NN-MR	55.29	**48.44**	52.03	20.00	open
2. Shot	1NN-MR	62.72	**56.64**	**54.55**	25.00	open

Labeling with **1NN** achieves very high accuracy in the "view 1". The "view 2" and combined views are more challenging. In "view 1" most of the frames have the figures correctly segmented, while in the "view 2" the segmentation often looses legs and arms of the player. Furthermore as the resolution decreases, the quality of the optic flow degrades. These factors make prediction problem on "view 2" very difficult. The combination of the views presents another challenge. We distinguish forehand and backhand shots, however forehand shot in one view

is similar to the backhand shot in the other view. This further degrades the classifier performance. Consistently with the structured dataset results, **1NN-R** performs worse than **1NN**, because the rejection problem is difficult.

1NN-M improves 1-6% from **1NN** on closed world. **1NN-MR** improves 0.5-4% performance on "view 2" but does not help on "view 1". In "view 1", some unseen activities are quite similar to some observed actions. For example, when the player stands and do nothing, we labelled as "unknown" motion and "unknown" shot. However it looks quite similar to "hop" motion and "backhand" shot because the camera looks from the back of the closer player. In this case, LMNN learns a metric for moving same-label inputs close together. Unfortunately, this transformation also collapses the unseen activities.

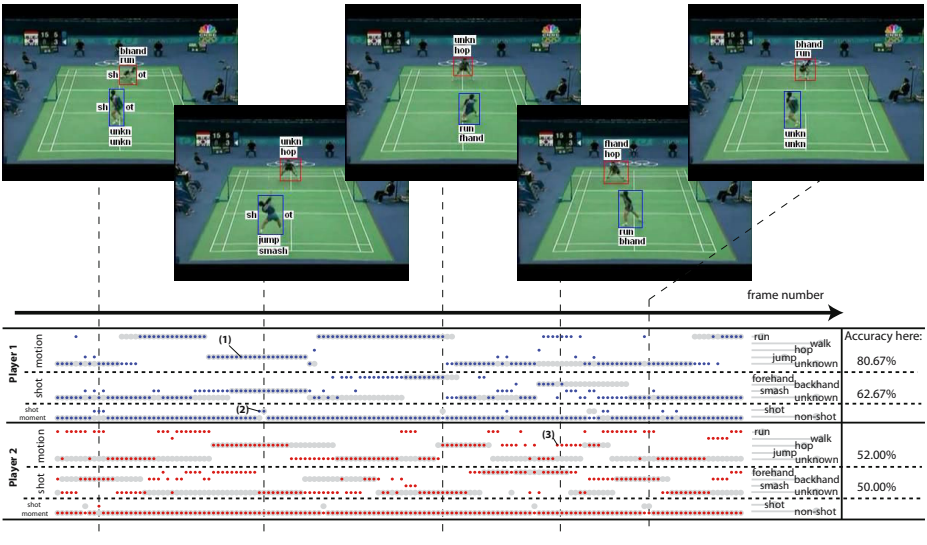


Fig. 6. Video labeling of Youtube sequences demonstrates the **1NN-R** in non-dataset environment. The figure is a snapshot from the video in the supplementary materials. Every frame is associated with one column. Every possible prediction is associated with one row. Gray dots denote the groundtruth which was labeled by hand. Blue dots are predictions for player 1 (close), red dots are predictions for player 2 (far). Predictions are grouped into 3 tasks (see table 3): 5 rows for motion type, 4 rows for the type of shot and 2 rows for shot instant prediction. The point marked (1) shows that at frame 1522 we predicted type of motion **jump** which is also labeled in the ground truth. The point marked (2) shows that at frame 1527 we predict that there is a shot. The ground truth marks the next frame. The point marked (3) shows that at frame 1592, we predict **hop**, while the groundtruth label is **unknown**. The accuracy numbers in the figure are computed for 150 frames shown in the figure.

As can be seen in table 5, our shot instant prediction accuracy works remarkably well: 47.9% of the predictions we make are within a distance of 2 from a labeled shot and 67.6% are within 5 frames. For comparison, a human observer

Table 5. Task 3. Shot prediction accuracy shows the percentage of the predicted shots that fall within the 5,7,9 and 11-frame windows around the groundtruth label shot frame. Note, that it is almost impossible for the annotator to localize the shot better than a 3-frame window (i.e. ±1).

View	**±2-shot**	**±3-shot**	**±4-shot**	**±5-shot**
View 1	59.15	69.01	69.01	70.42
View 2	43.08	55.38	63.08	67.69
2 Views	47.97	59.46	63.51	67.57

has uncertainty of 3 frames in localizing a shot instant, because the motions are fast and the contact is difficult to see. The average distance from predicted shots to ground truth shots is 7.3311 while the average distance between two consecutive shots in the ground truth is 51.5938.

For complete presentation of the results we rendered all predictions of our method in an accompanying video. Figure 6 shows a snapshot from this video. The figure has several more frames shown with detected shots. Datasets and source code are available at: http://vision.cs.uiuc.edu/projects/activity/.

7 Discussion

In this paper, we presented a metric learning-based approach for human activity recognition with the abilities to reject unseen actions and to learn with few training examples with high accuracy. The ability to reject unseen actions and to learn with few examples are very crucial when applying human activity recognition to real world applications.

At present we observe that human activity recognition is limited to a few action categories in the closed world assumption. How does activity recognition compare to object recognition in complexity? One hears estimates of $10^4 - 10^5$ of objects to be recognized. We know that the number of even primitive activities that people can name and learn is not limited to a hundred. There are hundreds of sports, martial arts, special skills, dances and rituals. Each of these has dozens of distinct specialized motions known to experts. This puts the number of available motions into tens and hundreds of thousands. The estimate is crude, but it suggests that activity recognition is not well served by datasets which have very small vocabularies. To expand the number we are actively looking at the real-world (e.g. Youtube) data. However the dominant issue seems to be the question of action vocabulary. For just one YouTube sequence we came up with 3 different learnable taxonomies. Building methods that can cope gracefully with activities that have not been seen before is the key to making applications feasible.

Acknowledgments. We would like to thank David Forsyth for giving us insightful discussions and comments. This work was supported in part by Vietnam Education Foundation, in part by the National Science Foundation under IIS - 0534837, and in part by the Office of Naval Research under N00014-01-1-0890

as part of the MURI program. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the VEF, the NSF or the Office of Naval Research.

References

1. Efros, A., Berg, A., Mori, G., Malik, J.: Recognizing action at a distance. In: ICCV, pp. 726–733 (2003)
2. Ramanan, D., Forsyth, D.: Automatic annotation of everyday movements. In: NIPS (2003)
3. Ikizler, N., Forsyth, D.: Searching video for complex activities with nite state models. In: CVPR (2007)
4. Howe, N.R., Leventon, M.E., Freeman, W.T.: Bayesian reconstruction of 3d human motion from single-camera video. In: Solla, S., Leen, T., Muller, K.R. (eds.) NIPS, pp. 820–826. MIT Press, Cambridge (2000)
5. Barron, C., Kakadiaris, I.: Estimating anthropometry and pose from a single uncalibrated image. CVIU 81(3), 269–284 (2001)
6. Taylor, C.: Reconstruction of articulated objects from point correspondences in a single uncalibrated image. CVIU 80(3), 349–363 (2000)
7. Forsyth, D., Arikan, O., Ikemoto, L., O'Brien, J., Ramanan, D.: Computational aspects of human motion i: tracking and animation. Foundations and Trends in Computer Graphics and Vision 1(2/3), 1–255 (2006)
8. Niyogi, S., Adelson, E.: Analyzing and recognizing walking gures in xyt. In: Media lab vision and modelling tr-223. MIT, Cambridge (1995)
9. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: ICCV, pp. 1395–1402 (2005)
10. Bobick, A., Davis, J.: The recognition of human movement using temporal templates. PAMI 23(3), 257–267 (2001)
11. Laptev, I., Lindeberg, T.: Space-time interest points. In: ICCV, pp. 432–439 (2003)
12. Laptev, I., Prez, P.: Retrieving actions in movies. In: ICCV (2007)
13. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
14. Wang, L., Suter, D.: Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model. In: CVPR (2007)
15. Wang, Y., Huang, K., Tan, T.: Human activity recognition based on r transform. In: Visual Surveillance (2007)
16. Ikizler, N., Duygulu, P.: Human action recognition using distribution of oriented rectangular patches. In: ICCV Workshops on Human Motion, pp. 271–284 (2007)
17. Arikan, O., Forsyth, D., O'Brien, J.: Motion synthesis from annotations. In: SIGGRAPH (2003)
18. Arikan, O., Forsyth, D.A.: Interactive motion generation from examples. In: Proceedings of the 29th annual conference on Computer graphics and interactive techniques, pp. 483–490. ACM Press, New York (2002)
19. Arikan, O.: Compression of motion capture databases. In: ACM Transactions on Graphics: Proc. SIGGRAPH 2006 (2006)
20. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning, with application to clustering with side-information. In: NIPS (2002)
21. Yang, L., Jin, R., Sukthankar, R., Liu, Y.: An effcient algorithm for local distance metric learning. AAAI, Menlo Park (2006)

22. Weinberger, K.Q., Blitzer, J., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. In: NIPS (2006)
23. Davis, J., Kulis, B., Jain, P., Sra, S., Dhillon, I.: Information-theoretic metric learning. In: ICML (2007)
24. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. PAMI 24(4), 509–522 (2002)
25. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stero vision. IJCAI, 121–130 (1981)
26. Achlioptas, D.: Database-friendly random projections. In: ACM Symp. on the Principles of Database Systems (2001)
27. Veeraraghavan, A., Chellappa, R., Roy-Chowdhury, A.: The function space of an activity. In: CVPR, pp. 959–968 (2006)
28. Weinland, D., Ronfard, R., Boyer, E.: Free viewpoint action recognition using motion history volumes. CVIU (2006)
29. Niebles, J., Fei-Fei, L.: A hierarchical model of shape and appearance for human action classification. In: CVPR, pp. 1–8 (2007)
30. Ali, S., Basharat, A., Shah, M.: Chaotic invariant for human action recognition. In: ICCV (2007)
31. Jhuang, H., Serre, T., Wolf, L., Poggio, T.: A biological inspired system for human action classification. In: ICCV (2007)
32. Scovanner, P., Ali, S., Shah, M.: A 3-dimensional sift descriptor and its application to action recognition. ACM Multimedia, 357–360 (2007)
33. Lv, F., Nevatia, R.: Single view human action recognition using key pose matching and viterbi path searching. In: CVPR (2007)

Shape Matching by Segmentation Averaging

Hongzhi Wang and John Oliensis

Stevens Institute of Technology
{hwang3,oliensis}@cs.stevens.edu

Abstract. We use segmentations to match images by shape. To address the unreliability of segmentations, we give a closed form approximation to an average over all segmentations. Our technique has many extensions, yielding new algorithms for tracking, object detection, segmentation, and edge-preserving smoothing. For segmentation, instead of a maximum a posteriori approach, we compute the "central" segmentation minimizing the average distance to all segmentations of an image. Our methods for segmentation and object detection perform competitively, and we also show promising results in tracking and edge–preserving smoothing.

1 Introduction

The shape of an object (as conveyed by edge curves) is among its most distinctive features, yet many methods for recognition/detection or tracking neglect it. One reason for this is that shape matchers confront a difficult global/local dilemma: local edges carry too little information for reliable matching, while globally the images have too much variability.

We classify shape matching strategies according to their shape representations. Methods representing shape *locally* [4,5] in terms of edgels confront a combinatorial explosion in the number of potential matches. *Global* methods have trouble extracting reliable global contours, and their matching suffers from occlusions and dropouts; also, global matching is hard because of the huge search space of possible deformations. Some recent shape-matching approaches [18,10] use grouped edge fragments as *intermediate* representations. These have more specificity and fewer potential matches than edgels yet occur often enough to survive occlusion and detection failures. But bottom–up grouping isn't reliable, so applying fragment groups to match general images is hard without top-down learning. As a result, [18,10] limit matching to specific objects or classes; they learn distinctive fragment groups for the given object(s) and match these representations. Other *semi–local* representations, e.g. SIFT [14] and HoG [8,6], achieve robustness to shape variation by weakening the shape descriptions, resorting to histograms instead of representing the exact boundary shapes.

We propose an approach to shape matching based on averaging over segmentations. The method combines the advantages of global and local approaches: it can match globally yet efficiently, and is robust to local variation yet remains sensitive to the detailed boundary shapes. Others [21,2] have used segmentation

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 562–575, 2008.

for recognition, but we differ in using it to represent shape. For general recognition, we can apply our method like SIFT as a (semi)local descriptor. Here, to demonstrate its power to match despite large variability, we apply it *globally*, in experiments on tracking and on localizing instances of an object class.

Our technique for averaging segmentations has implications beyond matching. Using it, we derive a segmentation method which gives competitive results on the Berkeley database. We also apply it for edge–preserving smoothing. Unlike previous ones, our smoothings are sensitive to *global* image structures.

2 Shape Matching: Motivations and Overview

Region based shape matching. To resolve the global/local dilemma, as a first step we avoid the local ambiguities of edge matching by instead matching *regions* [3]. As global features, regions have robustness to local shape distortions and occlusions, but they are difficult to extract reliably and can have complex shapes which are hard to represent or match. Since they have closed boundaries, they don't adapt easily for matching open image curves.

Segmentation matching. We next upgrade region matching to matching *segmentations*. This has several advantages: 1) By matching all regions at once, we gain robustness to the grouping failures for any one region; 2) Since segmentations are computed using global image information, they can localize the true edges more accurately than local edge–detection/grouping methods; 3) Segmentations can reveal global shape structures which are more distinctive than local features. This helps overcome difficulties caused by "hallucinated" boundaries; 4) An oversegmentation includes most of the strong curves and may be exploited for matching open as well as closed curves.

A typical segmentation includes both true and hallucinated boundaries. In matching segmentations, we need a similarity measure that detects the true matching boundaries while ignoring the hallucinations. The measure should be insensitive to small shifts and distortions in the boundaries.

A segmentation similarity measure. To achieve this, we propose a new similarity measure. It relates to the mutual information as adapted to segmentations; the basic concept is the *structure entropy* (SE), i.e., the entropy measuring a segmentation's complexity. A segmentation with many small segments has high structure entropy; large segments give low SE. (Note: we compute the structure entropy for individual segmentations, not for the probability distribution over segmentations.) We define the *structure mutual information* (SMI) between segmentations in terms of the joint segmentation obtained by superimposing the individual ones. When two segmentations have matching boundaries, the joint segmentation has large regions and low SE, so the SMI is high. For non–matching segmentations, the joint segmentation has smaller regions and higher SE, so the SMI is low. Fig.1(a) illustrates the idea. The precise definitions are below.

This similarity measure achieves the desired aims. For example, shifting one segmentation a small amount relative to another creates new small regions in

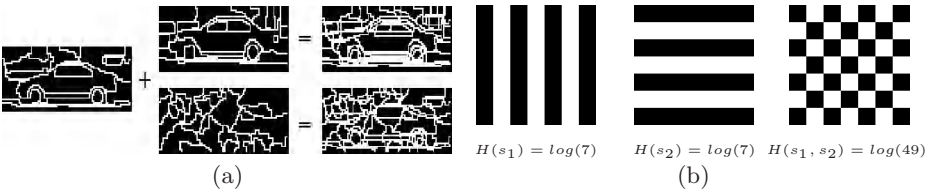


Fig. 1. (a). Segmentation based shape matching. Left 2 columns: segmentations to be matched. Right column: the joint segmentations. Two similar overlapping shapes (top) gives larger regions than overlapping dissimilar shapes (bottom); (b) 2 segmentations and their joint segmentation (right). The segmentations have VoI log(49) and MI 0.

the joint segmentation, but large segments continue to have large overlaps, so the joint SE remains low and the SMI remains high.

However, the approach still depends on the quality of the precomputed segmentations. In many cases, an oversegmentation includes enough of the true boundaries for good results, but it will also include fake boundaries, weakening shape accuracy. Our solution to this problem is our main theoretical contribution. Instead of using actual segmentations, we compute their average similarity, averaging over all possible segmentations for both images weighted by their probabilities. Indistinct boundaries also contribute to the average, so we get good matching even for unstructured images.

Segmentation averaging. On its face, averaging over all segmentations seems impossible. It is clear that we cannot do it exactly. We present an approximation which gives the average similarity between segmentations in closed form.

The main ideas in our approximation are as follows. First, we make a standard approximation to the segmentation probability distribution, representing it in terms of the local affinities between pixels. For example, one can consider normalized cuts (NCuts or NC) [23] as computing the maximum a posteriori (MAP) segmentation for a probability distribution defined by the affinities.

Our key realization is that we can compute the averaged structure entropy (and similarly the averaged SMI) separately in terms of each pixel's contribution. We show that a pixel's contribution to the SE is the geometric mean size of the segment containing it. The geometric mean is also hard to compute exactly, but we exploit a standard technique to approximate it in terms of the arithmetic mean. This approximation is a mild one: roughly, it originates from a Taylor expansion of the geometric mean around the arithmetic mean, and we show both theoretically and experimentally that the higher order terms in this expansion can be neglected. The following sections present the details of our approach.

3 Segmentation Similarity and Averaging

A segmentation similarity measure. We start by defining the structure entropy (SE). Given a segmentation s, we define a r.v. x which we consider as

ranging uniformly over all pixels. More precisely, we define the states of x by the segment labels in s, so the probability p_i for the i_{th} segment is the probability that a pixel lies in that segment, i.e., it is the area ratio of the segment to the whole image. For n segments, the structure entropy for s is

$$H(s) \equiv H(x) = -\sum_{i=1}^{n} p_i \log(p_i). \tag{1}$$

For two segmentations s, s' and corresponding r.v. x, y, the joint structure entropy $H(s, s') \equiv H(x, y)$ is the structure entropy of the joint segmentation. See Fig. 1(b). Let z be the r.v. for the joint segmentation. The possible states of z are the label pairs $(l_s, l_{s'})$ where, for each pixel, l_s and $l_{s'}$ give the containing segment from s and s' respectively. A pixel with labels l_s and $l_{s'}$ lies in the intersection of the corresponding segments.

Having defined the structure entropy, we define the mutual information (MI) of two segmentations in the standard way:

$$H(x; y) = H(x) + H(y) - H(x, y). \tag{2}$$

We call this the *structure mutual information* (SMI). We also define the *variation of information* (VoI) [17] as $\mathrm{V}(x, y) \equiv H(x, y) - H(x; y)$ or, equivalently,

$$\mathrm{V}(x, y) \equiv 2H(x, y) - H(x) - H(y). \tag{3}$$

[17] showed recently that the VoI gives a distance metric for clusterings. In our context, it gives a distance metric on segmentations[1].

3.1 The Averaged Structure Entropy

As described in Section 2, we match images by computing the average similarity of their segmentations. To do this, we need the average of the structure entropy over all possible segmentations. This section describes our main theoretical contribution: an approximation to this average.

Let F be the image. For a given segmentation s, the m_{th} pixel contributes

$$H^{(m)}(s) = -A_F^{-1} \log(A_F^{-1} A(s^{(m)})) \sim \log(A(s^{(m)})) \tag{4}$$

to the SE, where $s^{(m)}$ is the segment containing pixel m, $A(\cdot)$ gives its area, and $A_F \equiv A(F)$ is the total area of F. The SE is the sum of $H^{(m)}(s)$ over all pixels.

Let S denote the set of all possible segmentations of F. We define the *averaged structure entropy* ASE by

$$H_\omega(F) = \sum_{s' \in S} p(s'|F) H(s'), \tag{5}$$

[1] We derived our measures independently but later than Meila.

where $p(s'|F)$ is the conditional probability of the segmentation s' given F and $H(s')$ is the SE for s'. The contribution of the m_{th} pixel to the ASE $H_\omega(F)$ is

$$H_\omega^{(m)} = \sum_{s\in S} p(s|F)H^{(m)}(s) \sim \log[\prod_{s\in S} A(s^{(m)})^{p(s|F)}]. \tag{6}$$

Our key insight is: we can evaluate the ASE *without enumerating all segmentations* if we can compute the geometric mean segment size $G \equiv \prod_{s\in S} A(s^{(m)})^{p(s|F)}$.

The geometric mean is hard to compute. A common approximation [27] is $G \approx \mu - \sigma^2/2\mu$, where μ is the arithmetic mean and σ^2 is the variance. Then

$$G \approx E\{A(s^{(m)})\} - \mathrm{Var}\{A(s^{(m)})\}/(2E\{A(s^{(m)})\}). \tag{7}$$

(E is the expectation.) One can conveniently express the arithmetic mean and variance in terms of the *affinity matrix* M_F, defined by

$$M_F(m,n) = \sum_{s\in S} p(s|F)M_s(m,n), \tag{8}$$

where M_s is the *segmentation affinity matrix* for s with entries 1 (the pixels belong to the same segment) or 0. For an image with N pixels, $M_F \in \Re^{N\times N}$. Each entry measures the probability that the two pixels lie in the same segment.

For any pixels m, n, let $\chi_{(m,n)}$ be the indicator variable representing the event that the given pixels belong to the same segment. Then $E\{\chi_{(m,n)}\} = M_F(m,n)$. The m_{th} pixel's segment–size mean and variance are

$$\mathrm{E}\{A(s^{(m)})\} = E[\sum_n \chi_{(m,n)}] = \sum_n M_F(m,n) \tag{9}$$

$$\mathrm{Var}\{A(s^{(m)})\} = \sum_{k,l} cov(\chi_{(m,k)}, \chi_{(m,l)}) = \sum_{k,l} E[\chi_{(m,k)}, \chi_{(m,l)}] - E[\chi_{(m,k)}]E[\chi_{(m,l)}] \tag{10}$$

$$\leq \sum_{k,l} \min(M_F(m,k), M_F(m,l))(1 - \max(M_F(m,k), M_F(m,l))). \tag{11}$$

To apply in practice, we estimate M_F from local image properties, e.g.

$$M_F(m,n) \approx \begin{cases} exp(\frac{-(F_m - F_n)^2}{2\sigma^2}) & \text{if } d(m,n) \leq D; \\ 0 & \text{if } d(m,n) > D, \end{cases} \tag{12}$$

where F_m is the intensity, $d(m,n)$ the distance between pixels m, n, and σ the standard deviation of intensity within a segment. (12) embodies the principle that nearby pixels with similar intensity are more likely to group than distant pixels with different intensities. Other cues, e.g., texture or the presence of an intervening edge [15], could be used as well.

We simplify further by treating the pairwise probabilities (that two pixels belong to the same segment) as independent. Then we have:

$$p\left(s|F\right) \approx \frac{1}{Z} \prod_{m\geq n} M_F\left(m,n\right)^{M_s(m,n)} \left(1 - M_F\left(m,n\right)\right)^{1-M_s(m,n)},$$

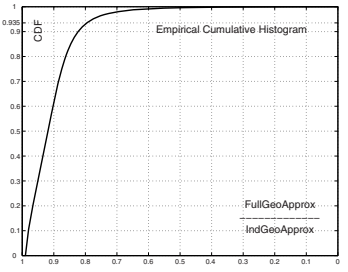


Fig. 2. Empirical distribution shows small impact of the independence assumption

where Z is a normalization constant. As discussed in Section 2, this assumption is a common one and underlies segmentation algorithms such as NC. Using it,

$$\mathrm{Var}\{A(s^{(m)})\} \approx \sum_n M_F(m,n)(1 - M_F(m,n)). \tag{13}$$

(9) and (13) imply $\frac{Var\{A(s^{(m)})\}}{2E\{A(s^{(m)})\}} < 0.5$. Since meaningful segments are sizable, we expect their mean segment size $E\{A(s^{(m)})\} = \mu \gg 0.5$, implying that the arithmetic mean approximates the geometric mean well. We use this approximation in all our experiments, taking $H_\omega(F) \approx \sum_m \log\left(\sum_n M_F(m,n)\right)$.

Validation of pairwise independence. For us the pairwise–independent approximation is especially appropriate, since we only keep the affinities of nearby pixels (only nearby affinities can be reliably estimated) and these are mostly near 1 with small covariances, see (11). As a check, Fig. 2 shows the empirical distribution of $\frac{\bar{G}_{Full}}{G_{Ind}}$ over real images, where $\bar{G}_{Full}$ is a lower bound on the geometric mean computed from (11) without the independence assumption, and G_{Ind} is the mean computed assuming independence.

The images are the Berkeley segmentation training data [16] (200 images here resized to 80×120). For any image, each pixel gives a sample of the lower bound. We computed the affinity matrices using the Gaussian (12) with deviation $\sigma = 10$ (intensity range [0,255]) and $D = 5$. For over 93.5% of the pixels, the ratio lower bound is above 0.8, implying that the independence assumption has little effect in practice. Note that Fig. 2 plots a lower bound on the ratio, so the ratio itself will have an even better empirical distribution.

A modification. A potential issue is that estimates of the pairwise probabilities in M_F are reliable only over a small region (D in (12) should be small). Thus, we average the structure entropy with respect to the neighborhood size instead of the full image size, redefining the ASE as:

$$H_\omega = -\frac{1}{A_F} \sum_m \log\left(\frac{E\{A(s^{(m)})\}}{A_m}\right) \tag{14}$$

where A_m is the neighborhood size for the m_{th} pixel.

3.2 Shape Similarity from Averaging Segmentations

We compare the shapes in two images by computing the average distance between the image segmentations. Recall the definition (3) of VoI, which gives a metric on segmentations. The average value of this metric, averaged over all possible segmentations for two images F_1 and F_2, is $V_\omega(F_1, F_2) \equiv 2H_\omega(F_1, F_2) - H_\omega(F_1) - H_\omega(F_2)$, where the $H_\omega(F_a)$ are the ASE for the two images, and

$$H_\omega(F_1, F_2) = \sum_{s_1} \sum_{s_2} p(s_1|F_1)p(s_2|F_2)H(s_1, s_2) \tag{15}$$

is the ASE of the joint segmentations for the two images. As before, we approximate $H_\omega(F_1, F_2) \approx \sum_m \log\left(\sum_n M_{F_1F_2}(m, n)\right)$; the joint affinity matrix is:

$$M_{F_1F_2}(m, n) = M_{F_1}(m, n)M_{F_2}(m, n). \tag{16}$$

The VoI $\mathrm{V}(s, s')$ has the joint structure entropy $H(s, s')$ as an upper bound. When one searches for the most similar segmentation to a given segmentation or image, this biases the result toward segmentations of low complexity. To compensate for the bias, we normalize our averaged distance, using $\Delta(F_1, F_2) \equiv \frac{V_\omega(F_1,F_2)}{H_\omega(F_1,F_2)}$ as our measure for image comparisons. Note that $0 \leq \Delta \leq 1$, where $\Delta = 1$ implies that the images are very dissimilar. Our approximations give

$$\Delta(F_1, F_2) \approx 2 - \frac{\sum_m \log\left(\sum_{nn'} M_{F_1}(m, n)M_{F_1}(m, n')\right)}{\sum_m \log\left(\sum_n M_{F_1}(m, n)M_{F_2}(m, n)\right)} = 2 - \frac{\sum_m \log\left(\bar{A}_{1m}\bar{A}_{2m}\right)}{\sum_m \log\left(\bar{A}_{\mathrm{J}m}\right)}, \tag{17}$$

where $\bar{A}_{am}$ is the mean size of the containing segment for the individual or joint segmentation. Roughly, Δ measures the statistical independence of the segment sizes in the two images. Sec. 5 applies Δ in tracking and detection experiments.

Comparison to other measures. The Probability Rand (PR) index [26] can also be considered a similarity metric based on affinity matrices:

$$PR(F_1, F_2) \propto \sum_m [2\sum_n M_{F_1F_2}(m, n) - \sum_n M_{F_1}(m, n) - \sum_n M_{F_2}(m, n)] \tag{18}$$

Note that PR sums the affinities separately, ignoring all spatial interactions between nearby pixels. Another affinity-based measure, the similarity template [25] (ST), includes the spatial interactions for each image separately but not for the joint affinity. Our metric does include spatial interactions. Sec. 5 shows experimental comparisons of our approach with PR and ST.

4 Application to Segmentation and Smoothing

Segmentation algorithms such as NC can be considered as computing the MAP segmentation. The wide divergence in segmentations found by different methods, and the imbalance between small/large segments, suggest that the probability of segmentations has a broad asymmetric peak. For any such r.v., the mean is the estimator with least variance and usually superior to the MAP. Can we use our averaging technique to approximate the mean segmentation?

The central segmentation. Since the mean has least variance, we can compute it for a r.v. x as $\bar{x} = \text{argmin}_x \sum_y p(y)|y-x|^2$. Recall that $V(s,s')$ gives a metric for segmentations. Given an image F, we define its *central segmentation*

$$\hat{s} \equiv \text{argmin}_s \sum_{s' \in S} p(s'|F)\text{V}(s',s) \equiv \text{argmin}_s V_\omega(s,F). \tag{19}$$

$\hat{s}$ is "central" in that it minimizes the average distance V to all segmentations of F. It is the mean segmentation with respect to the distance metric $\sqrt{V}$. We choose $\hat{s}$ partly for convenience, since we already approximated V_ω; also, we expect *better* segmentations using $\sqrt{V}$ and $\hat{s}$ than for the metric V.

Our reasons for this expectation are as follows. A typical image has many qualitatively different yet plausible segmentations, implying $p(s|F)$ has many large peaks. Averaging over all s combines qualitatively different segmentations, whereas we want the center of the dominant peak. Using the metric $\sqrt{V}$ for the average gives a robust estimate with more resistance to outlier segmentations.

Note that [11] also segments by averaging over segmentations. Differences include: [11] computes the mean affinity matrix, not the segmentation directly, and averages by *sampling*. Our result is closed form and applies more generally, e.g., to matching; [11] focuses just on segmentation.

We again normalize the averaged distance, redefining $\hat{s} \equiv \text{argmin}_s \frac{V_\omega(s,F)}{H_\omega(s,F)} \equiv \text{argmin}_s \Delta(s,F)$, where $H_\omega(s,F) \equiv \sum_{s' \in S} p(s'|F)H(s,s')$.

Fig. 3. Segmentation results for naive greedy merging

Segmentation algorithms. We use the simple affinity matrix M_F of (12) to compute $V_\omega(s,F)$, $H_\omega(s,F)$ in all our experiments. We compute $\hat{s}$ by iteratively minimizing $\Delta(s,F)$. We used two iterations; the first is fast greedy merging (GM). We start with each pixel as a segment, then merge neighbor segments if this decreases $\Delta(s,F)$. Because M_F is nonzero just over small neighborhoods, we can compute the merged segmentation with linear cost $O(NW)$, where W is the neighborhood size determined by D in (12). We used $D = 5$ for GM, giving $W = 121$. To avoid local minima, one can repeat the merge several times, starting with a small σ in M_F (e.g., 1) and then gradually increasing it to a specified value. Fig.3 shows this naive method can give excellent segmentations, indicating the robustness of our segmentation criterion $\Delta(s,F)$.

Our second method uses gradient descent. We represent a segmentation by *real-valued* labels s_m at each pixel m, with neighboring pixels in the same segment *iff* their labels differ by < 1. Letting I denote the indicator function,

$$H(s) = -\sum_m \log \frac{\sum_n M_s(m,n)}{A_m} = -\sum_m \log \frac{\sum_n I(|s_n - s_m| < 1)}{A_m} \tag{20}$$

$$H_\omega(s,F) = -\sum_m \log \frac{\sum_n I(|s_n - s_m| < 1) M_F(m,n)}{A_m}. \tag{21}$$

We initialize s to the original intensity image. Since we require smooth gradients, we approximate the derivative of $I(|s_m - s_n| < 1)$ by a smooth function.[2] Using a smoother approximation speeds up convergence and extends the search for $\hat{s}$ over a larger range. Note that we do *not* change our criterion $\Delta(s,F)$ for a good segmentation. Though our approximations may cause $\Delta(s,F)$ to increase after some iterations, at the end the algorithm outputs the s giving the least $\Delta(s,F)$. For even faster convergence, we add a "force" term $\gamma \frac{\partial H_\omega(s,F)}{\partial s_m}$ to the gradient, where we set $\gamma = 0.5$. Again, adding this term widens the search but does not affect our criterion $\Delta(s,F)$. The force acts as a regularization that focuses the search on simpler segmentations, helping overcome local minimum. With these changes, we usually get convergence in a few hundred iterations. We always ran for 500 iterations in our experiments, which takes a few seconds on a 3.2G Hz AMD for images with 10000 pixels (code available on our web page).

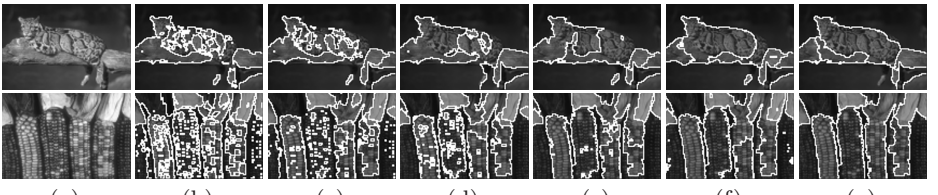


Fig. 4. a). Input images (80×120); b)-g). Segmentations after 1 - 6 optimization rounds. After each round, we update standard deviations via (22) to encourage merging of small segments. 1_{st} row: $(D,\sigma,\alpha,t) = (2,50,0.25,20)$;$2_{nd}$ row: $(D,\sigma,\alpha,t) = (2,30,0.5,20)$.

Initially, we set σ in M_F to a constant. This doesn't allow for changes across the image [9], as occur especially in textured regions. To deal with such changes, we adapt σ locally based on the current segmentation:

$$\sigma_{(m,n)} \longleftarrow \sigma_{(m,n)}(1 + e^{-\frac{A(s^{(m)})}{t}})(1 + e^{-\frac{A(s^{(n)})}{t}}), \tag{22}$$

where the parameter t acts as a threshold that encourages segments smaller than t pixels to merge into their neighbors, with little effect on large segments. Finally, we repeat the whole round of gradient descent/σ-update. The results stop changing after a few rounds; we ran 6 rounds in all our segmentation experiments. As Fig. 4 shows, updating σ highlights the large salient structures, adding some stability in textured regions. (We don't model texture explicitly as in [22], so we cannot expect competitive performance in texture segmentation.)

[2] $\max(1, |s_m - s_n|)^{-\alpha} \text{sign}(s_m - s_n)$. We used $\alpha = 0.25, 0.5$.

Smoothing. Segmentation relates closely to edge–preserving smoothing; in fact, one can consider it as a piecewise constant smoother. We implement edge–preserving smoothing by finding the "most similar" image to the original image F according to our criterion Δ. The algorithm is steepest descent, similar to the second segmentation procedure (but without the derivative approximation or extra force), except we optimize Δ over images instead of segmentations.

The most similar image F_{sim} does *not* equal the original. Instead, one can show that it has a segmentation probability distribution which clusters around $\hat{s}$, the central segmentation of F. As a result, F_{sim} agrees with the boundaries of F and varies smoothly over its non–boundary regions (to discourage unlikely segmentations). Unlike previous methods based on local computations, our approach smooths according to the image's *global* optimal structures. By averaging over probabilities, it can adjust the smoothing according to boundary strength and smooth *across boundaries*, not just within segments. Fig. 5 shows our algorithm gives appealing smoothings which preserve accurate contours.

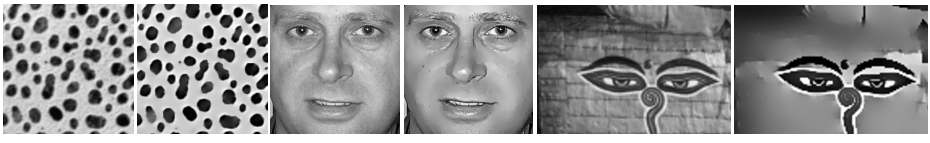


Fig. 5. Left: the original images; Right: results after 30 iterations of smoothing. $D = 2$. The σ used are 60,20,60 respectively.

5 Experiments: Segmentation, Detection, and Tracking

Image segmentation. Fig. 6 shows sample segmentations by our methods compared to the best ones from EDISON mean-shift (MS) [7]. On images with faint structures and large local ambiguity, MS often gives less accurate boundaries. For a quantitative comparison, we use the Berkeley segmentation test set [16] with each gray image resized to 80×120. Since the standard boundary-consistency criterion isn't optimal for evaluating segmentations, we use region consistency as our criterion. We consider multiple human-labeled segmentations as giving a probability distribution for the "ground truth," and evaluate segmentations by their averaged distance from this distribution using $\Delta(s_1, s_2) = V(s_1, s_2)/H(s_1, s_2)$. (Note this is *not* the $\Delta(s, F)$ minimized by our algorithm!) We applied MS using default parameters and intensity cues only. Our method used $D = 2, \alpha = 0.5,\ t = 2$. We also tested NC [23]. For each method, we obtain 3 segmentations per image by: varying the minimum region size as 5, 10, and 50 (MS); choosing $\sigma = 40, 60, 100$ (ours); specifying 60,40,20 segments (NC). The parameters are chosen s.t. the segmentations for different methods contain similar number of segments. Table 1 shows our method does best.

Table 1. The mean average distance to "ground truth" segmentations

average segment #	Ours	MS	NC
20	0.644	0.653	0.662
40	0.632	0.654	0.685
60	0.629	0.657	0.699

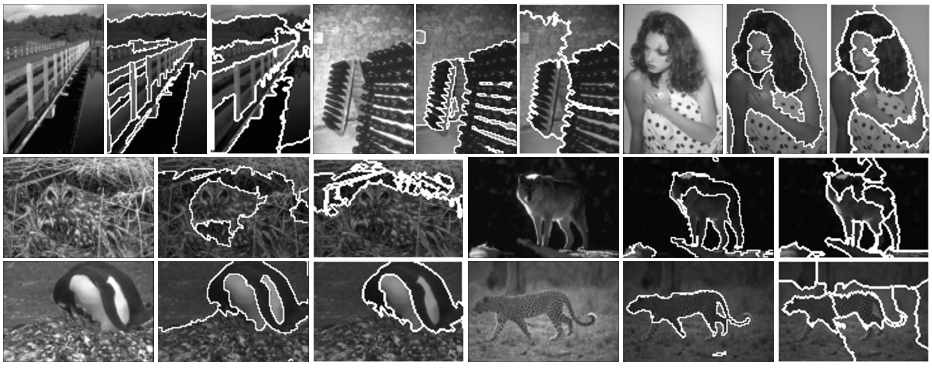


Fig. 6. Sample segmentation results. 1_{st}: Input images, size 120×80 or 80×120; 2_{nd}: segmentations by our approach; 3_{rd}: segmentations by mean shift (EDISON).

Detection. The "bag of features" (BoF) approach to recognition compares images according to the appearance (textures) of small patches. To cope with global variability, it takes a resolutely local approach, neglecting spatial layout almost completely. More recent work uses some layout information [12], or local shape descriptors [18,10,24], or a combination of local appearance and shape [19], and gives improved performance especially for object detection (i.e., localization). As discussed in the introduction, the local–shape approaches rely on supervised learning: to match an image, they must first build a descriptor of its contents from training images.

Our approach can match shapes for general images. It complements the local–appearance methods, and we could combine the approaches by treating ours as another semi-local descriptor (but for shape). This would be the appropriate strategy for recognition of general non–rigid objects.

Here, we concentrate on testing our method, applying it for detection on its own. To demonstrate its robustness against global variability, we apply it as a *global* descriptor: We detect an instance by measuring test images against a template containing the *whole* object plus context. We use minimal learning, detecting objects by thresholding our similarity measure with non-maximum suppression [1]. We ran detection experiments on the UIUC car side-views and the CalTech car rear-view and face data. The UIUC data contains 550 positive exemplars. We used their average affinity matrix as the car-side template, detecting cars in test images by thresholding the normalized distance from the

template. For the face and car rear-view data, we manually cut 10 positive examples from the first 10 test images and detected by thresholding the average distance from these exemplars. Templates were scanned across the test images.

Instead of using (12), we estimated the affinities separately for each image based on its statistics. Given image F, let h_r^F be the empirical histogram of the absolute intensity difference between pixels at a relative distance r. To get the affinities, we normalize so that $h_r^F \in [0,1]$ and average over r:

$$h^F(i) = D^{-1} \sum_{r=1:D} h_r^F(i) \tag{23}$$

The normalization gives higher weight to the closer pixels. The choice for the cut off D should reflect the scale of the object. For the car-side, car-rear, and face data, we used $D = 2, 7, 5$ respectively.

Table 2. Detection performances: equal error rates (percentage)

	Ours	[25]	[26]	[18]	[24]	[19]
Faces	98.1	93	92	96.4	97.2	99
Crear	100	98.2	86	97.7	98.2	100
Cside	90	–	–	85	–	93.8

As expected, our method gives better results than [25,26] (see Table 2). Our results are also better than [18,24], which learn local shape descriptors from training images, assuming, as we do, that the object is delineated by a bounding box. [19] does better on car-side and slightly better on faces, but unlike us uses both shape and appearance. Considering all reported results on these data, our performance on faces and car-rear views is close to the state of the art with many fewer exemplars, with slightly worse results for the car-side data. The latter contains images with significant occlusion, so our global matching strategy cannot compete with approaches based on local features. The best result for the car–side data is 97.5% [13], but this algorithm unlike ours has a verification stage; without this stage, it gives performance similar to ours.

Tracking. We next present a simple application of our match measure to tracking. Many shape–based trackers use global active contours; these require good initializations around the object of interest. Other approaches weaken the shape representation to make it robust to shape distortion, for instance representing the object in terms of histograms over gradients, e.g. [8].

Our approach tracks an object by its detailed curve shapes. As a shape tracker, it can localize the object more accurately than histogram–based ("blob") trackers. Since our matching measure has a built-in robustness to shape distortions, we can implement tracking essentially as simple template matching.

The user selects a window around an object in the starting image, and the algorithm tracks by moving the window to the best shape match in each new image (no motion, background information, or learning). Currently, we use brute

Fig. 7. Tracking results. Bottom corner of each frame shows expected segment size for each pixel of current template: dark pixels lie near boundaries. Our brute-force search Matlab routine takes a few sec/frame with a 3200HZ AMD.

force search to find the best location; using iterative gradient ascent would give a faster algorithm. To handle occlusion, we include history into the current window representation, updating its affinity matrix description by $M_F^{(n)} = (19M_F^{(n-1)} + M_{F(n-1)})/20$, where $M_{F(n-1)}$ is the affinity matrix for image $n-1$ alone, and we use $M_F^{(n)}$ for matching. Our results (e.g., Fig. 7) on a PETS 2007 sequence, and on outdoor and indoor sequences from [20], show the method's robustness to occlusion, deformation, camera motion, and changes in illumination, scale and pose. Our tracking is near perfect; for complete results see our web page.

6 Conclusion

We use image segmentations for shape matching. Our approach can match curves without correspondence, over large-scale image regions, and with good robustness to local shape variations and occlusion. It can exploit global shape structures, which are more distinctive than local features. To address the unreliability of image segmentations, we describe a closed form approximation to an average over all segmentations. Our approach has many extensions, yielding algorithms for tracking, segmentation, and edge–preserving smoothing. In addition, we can apply our approach for the objective evaluation of segmentation algorithms, and for comparing computed segmentations to multiple "ground–truth" segmentation produced by humans. Finally, since our approach compares signals based on their internal structure, it can match signals from different modalities, e.g. images from different frequency bands, or visual images matched to sonar data.

References

1. Agarwal, S., Roth, D.: Learning a sparse representation for object detection. In: Tistarelli, M., Bigun, J., Jain, A.K. (eds.) ECCV 2002. LNCS, vol. 2359, pp. 113–127. Springer, Heidelberg (2002)
2. Ahuja, N., Todorovic, S.: Learning the taxonomy and models of categories present in arbitrary images. ICCV (2007)
3. Basri, R., Jacobs, D.: Recognition using region correspondences. In: IJCV (1997)
4. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. PAMI 24(4), 509–522 (2002)
5. Berg, A., Malik, J.: Geometric blur for template matching. In: CVPR (2001)
6. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: CIRV (2007)

7. Comaniciu, D., Meer, P.: Mean shift: a robust approach towards feature space analysis. IEEE Trans. PAMI 24(5), 603–619 (2002)
8. Danal, N., Triggs, B.: k Histograms of oriented gradients for human detection. In: CVPR (2005)
9. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. IJCV 59(2) (2004)
10. Ferrari, V., Jurie, F., Schmid, C.: Accurate object detection with deformable shape models learnt from images. In: CVPR (2007)
11. Gdalyahu, Y., Weinshall, D., Werman, M.: Self organization in vision: stochastic clustering for image segmentation, perceptual grouping, and image database. IEEE Trans. PAMI 23(10), 1053–1074 (2001)
12. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene catergories. In: CVPR (2006)
13. Leibe, B., Leonardis, A., Schiele, B.: Combined object catergorization and segmentation with an implicit shape model. In: ECCV workshop on SLCV (2004)
14. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV (2004)
15. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and texture analysis for image segmentation. IJCV (2001)
16. Martin, D.R., Fowlkes, C.C., Tal, D., Malik, J.: A database of human segmented natural images and its applications to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV (2001)
17. Meila, M.: Comparing clusterings by the variation of information. In: COLT (2003)
18. Opelt, A., Pinz, A., Zisserman, A.: A boundary-fragment-model for object detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 575–588. Springer, Heidelberg (2006)
19. Opelt, A., Pinz, A., Zisserman, A.: Fusing shape and appearance information for object catergory detection. In: BMVC (2006)
20. Ross, D., Lim, J., Lin, R.-S., Yang, M.-H.: Incremental learning for robust visual tracking. In: IJCV (2007)
21. Russell, B., Efros, A., Sivic, J., Freeman, W., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: CVPR (2006)
22. Sharon, E., Galun, M., Sharon, D., Basri, R., Brandt, A.: Hierarchy and adaptivity in segmenting visual scenes. Nature (2006)
23. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. PAMI 22(8), 888–905 (2000), `http://www.cis.upenn.edu/~jshi/software/`
24. Shotton, J., Blake, A., Cipolla, R.: Multi-scale categorical object recognition using contour fragments. IEEE Transactions on PAMI (2008)
25. Stauffer, C., Grimson, E.: Similarity templates for detection and recognition. In: CVPR (2001)
26. Unnikrishnan, R., Pantofaru, C., Hebert, M.: Toward objective evaluation of image segmentation algorithms. IEEE Trans. PAMI 29(6), 929–944 (2007)
27. Young, W.E., Trent, R.H.: Geometric mean approximation of individual security and portfolio performance. J. Finan. Quant. Anal. 4, 179–199 (1969)

Search Space Reduction for MRF Stereo

Liang Wang[1], Hailin Jin[2], and Ruigang Yang[1]

[1] Center for Visualization and Virtual Environments, University of Kentucky, USA
[2] Advanced Technology Labs, Adobe Systems Incorporated, San Jose, CA, USA

Abstract. We present an algorithm to reduce per-pixel search ranges for Markov Random Fields-based stereo algorithms. Our algorithm is based on the intuitions that reliably matched pixels need less regularization in the energy minimization and neighboring pixels should have similar disparity search ranges if their pixel values are similar. We propose a novel bi-labeling process to classify reliable and unreliable pixels that incorporate left-right consistency checks. We then propagate the reliable disparities into unreliable regions to form a complete disparity map and construct per-pixel search ranges based on the difference between the disparity map after propagation and the one computed from a winner-take-all method. Experimental results evaluated on the Middlebury stereo benchmark show our proposed algorithm is able to achieve 77% average reduction rate while preserving satisfactory accuracy.

1 Introduction

Stereo reconstruction has been one of the most investigated topics in Computer Vision for decades. Recently there have been great advances which are based on global energy minimizations. According to the Middlebury benchmark [1], most top performers are based on the Markov Random Field (MRF) formulation that solves stereo matching by minimizing certain cost functions. The two most popular algorithms for approximate inference on Markov Random Fields are Belief Propagation (BP) [2] and Graph Cuts (GC) [3,4]. Although being able to produce top-ranking results, all the algorithms based on the MRF formulation suffer significantly from heavy memory requirements and high computational complexities for large images and disparity search ranges. For instance, the memory required by BP to store all the messages scales on the order of $O(|I| \times |S| \times N)$, where I is the image, S is the disparity range and N is the size of the neighborhood system. For a 1-megapixel image with 200 disparities, a BP-based algorithm would need $3.2GB$ RAM just to only store all the single-precision floating-point messages using the standard 4-connected neighborhood system. GC is less memory consuming. However, as suggested in [5], most min-cut/max-flow algorithms behave nonlinearly (closer to quadratic) to the number of disparities. As a result, when the number of disparities goes up, which occurs naturally in high-resolution or wide-baseline stereo images, neither BP nor GC is practical to run on ordinary computers.

To address the above issue, we present in this paper a novel algorithm for reducing disparity ranges for MRF-based stereo. Regular MRF solver can be

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 576–588, 2008.

applied on this reduced label space, which leads to less memory usage and lower computational complexity. Our algorithm springs from the following two intuitions: First reliably matched pixels, for instance the *ground control points* (*GCPs*) [6], need less regularization in the global optimization step; Second, neighboring pixels should have similar disparity search ranges if their pixel values are similar. In more details, our algorithm starts from classifying each pixel as either stable or unstable based on its local cost distribution together with a left-right consistency check [7]. We formulate this bi-labeling process as a maximum a posterior MRF (MAP-MRF) problem which can be efficiently solved using either BP or GC. We then propagate the stable matches to the rest of the image following our second intuition to produce a complete disparity map. Inspired by [8], we formulate this propagation problem using a quadratic cost function which are solved using standard techniques. The complete disparity map from the propagation stage and the one computed in the local winner-take-all method guide the selection of the final per-pixel search ranges. We evaluated our algorithm on the Middlebury data sets with ground-truth and found that our algorithm is able to achieve 77% average reduction rate while preserving satisfactory accuracy.

1.1 Relation to Previous Work

Our work is most related to [9], which aims at reducing search space for GC based stereo algorithms. The authors proposed a reduction strategy based on window matching. The basic idea is to collect disparities from other pixels within a spacial window to construct the search range for the center pixel. They did not take reliable matching into consideration and the reduced search ranges may contain redundancies, especially for pixels near depth discontinuities. There was no explicit number on the reduced label space and an average 2.8 time speed-up was reported. Given the typical nonlinear complexity of GC, our search space reduction rate around 75% would have resulted in a speedup factor over 4.

This work also relates to a recent paper by Yu et al. [10], who studied the feasibility of applying compression techniques to the messages in the BP algorithm to improve the memory efficiency. Instead of treating the messages as generic data, we investigate the problem from a more domain-specific perspective by finding a plausible subset of search space derived from the image data.

By using reliably matched pixels as a starting point, our work also falls in the category of semi-dense stereo matching [11,12,13,14,15]. We emphasize that our reliable matching extraction algorithm is designed to address a different problem from these methods. Instead of finding the "most likely" disparity assignment for a given pixel, we decide whether its disparity given by the winner-take-all approach is reliable or not. Hence what we try to solve is a binary segmentation problem whose complexity does not increase with respect to disparity search ranges. In addition our algorithm has many distinct features and advantages. Firstly, our reliability measure is based on probabilistic models studied from a set of training stereo images with ground-truth disparities and thereby the most sensitive parameters are found automatically. Secondly, with a well-designed

energy function the found $GCPs$ are almost free of outliers. What is more, the global optimization formulation allows existing MRF stereo solvers to be employed directly to the stable matching extraction.

2 Search Space Reduction

The general framework for MRF-based stereo can be defined as follows: Let I be the reference image and S be a finite set of disparity candidates. A disparity function f assigns each pixel $p \in I$ a disparity value $f_p \in S$. The quality of the labeling function is given by an energy function of the form:

$$E(f) = \sum_{p \in I} C_p(f_p) + \sum_{(p,q) \in \xi} P(f_p, f_q). \tag{1}$$

where the first term, the data term, measures how well the labeling function agrees with the image data, while the second term, the smoothness term, encourages neighboring pixels to have similar disparity assignments based on the assumption that the scene is locally smooth. The optimal labeling function for the energy $E(f)$ can be approximately solved using either BP or GC. For more details, we refer the reader to [16].

This paper addresses the following problem within the framework: How to construct a non-empty subset S_p from S for each pixel such that the optimal labeling function based on $\{S_p\}$, where $\{S_p\} \doteq \cup S_p$, is as close as possible to the optimal labeling function based on S. We intuitively define the reduction rate as:

$$\Re = 1 - \frac{\sum_{p \in I} |S_p|}{|I| \cdot |S|}. \tag{2}$$

2.1 Matching Cost Computation

We first compute the per-pixel matching cost using the algorithm proposed by Birchfield and Tomasi [17]. We then aggregate the per-pixel matching cost over a spacial window. To avoid the "fattening" artifacts and be efficient, we adopt the two-pass adaptive algorithm proposed by [18] which is an approximation of the bilateral filtering of [19] which has shown to be remarkably effective for obtaining high quality disparity maps [18,20]. This approximation reduces the per pixel complexity from $O(\ell^2)$ to $O(\ell)$, where ℓ is the side length of the square window. Our implementation shows, although the disparity map from winner-take-all is slightly noisier compared to the result from full 2D aggregation, the processing time for Tsukuba image is only 1.2 seconds with approximation.

We emphasize that the adaptive aggregation is introduced to disambiguate the inaccurate pixel-wise matching cost for the search space reduction purpose. In Fig. 1 we show the winner-take-all disparity maps and our approximate two-pass aggregation in the first two columns. The error percentage in non-occluded areas are 69.4% and 16.6% respectively based on the Middlebury evaluation system [1]. We further demonstrate the disparity maps from GC using the corresponding

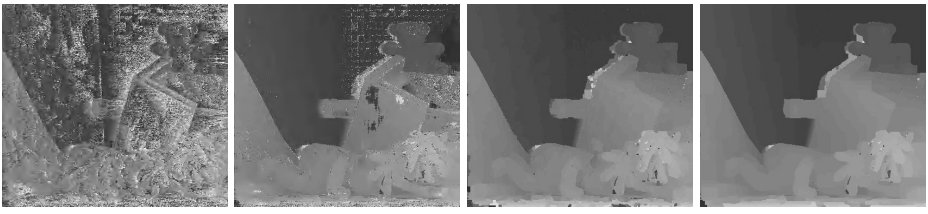


Fig. 1. From left to right: disparity map computed before ($e\% = 69.4$) and after ($e\% = 16.6$) adaptive cost aggregation using winner-take-all; the disparity map computed using Graph Cuts from the pixel-wise matching cost ($e\% = 9.25$); the disparity map computed using Graph Cuts after cost aggregation ($e\% = 6.81$)

matching cost as the data term in the third and fourth columns. While the results are visually similar, our aggregation step not only reduces the error rate from 9.25% to 6.81% but also brings GC's converging time from 27.8 seconds down to 24.2 seconds. It can be seen that MRF stereo can benefit from more reliable matching cost in terms of both quality and efficiency.

2.2 Stable Matching Extraction

Inspired by the fact that a majority of correct disparities can be estimated from their local costs after aggregation, we propose to reduce search space by locating those stably matched pixels and limit their disparity ranges substantially. There are existing algorithms designed to find unambiguous disparity assignments and derive semi-dense disparity maps. For instance, [11] uses dynamic programming to selectively assign disparities to pixels whose reliability exceed a given threshold; [12] utilizes an ordering constraint and uniqueness constraint together with some confidence measure to detect unambiguous component. Since these approaches need to explore the entire 3D cost volume, their computational complexities increase with respect to the search range.

To efficiently find the unambiguously matched pixels, we formulate a stable matching extraction as a binary segmentation problem. More specifically, given a disparity map D computed from winner-take-all, a function β assigns each pixel p with a label β_p: $\beta_p = 1$ if p's disparity estimate $D(p)$ is close enough to its true disparity and $\beta_p = 0$ otherwise. Our objective is to find the optimal labeling function that minimizes the following energy:

$$E(\beta) = \sum_{p \in I} U_p(\beta_p) + \sum_{(p,q) \in \xi} V(\beta_p, \beta_q). \tag{3}$$

Equation 3 is similar in form to equation 1 where the first term measures how well the configuration from β fits given matching data and the second term encodes a smoothness assumption on the binary labels.

Data Term. Binocular stereo matching is ill-posed inherently. Matching ambiguities can originate from many factors, among which insufficient signal-to-noise

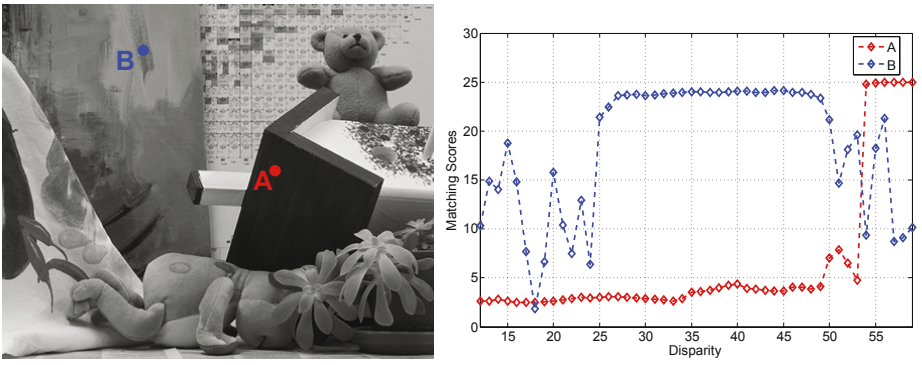


Fig. 2. Left: an image from the Teddy data set. Two points, A and B, are labeled in color. Point A belongs to a weakly texture region and point B belongs to an area with rich texture. Right: the cost profile of points A and B. Note that B's cost distribution has a clear global minimum while A's cost distribution is relatively flat.

ratio (SNR) in weakly textured regions and missing data in occlusion areas are the two most dominant ones [12]. To provide some insight into the behavior of matching costs, we select two scene points from the Teddy data set (shown in Fig. 2). The point A belongs to a textureless region while point B lies in an area with rich texture. On the right we plot their cost distributions respectively. We can see that B has a distinct global minimum at disparity 18. A's cost profile, however, is flat from disparity 0 to 35 and there is no distinctive global minimum. By referring to the ground-truth disparity map, B's true disparity is exactly 18 which is consistent with our estimate. A's disparity is incorrectly estimated at disparity 17 since $C_A(17) = 2.498$, which is slightly better than the cost 2.623 given by A's true disparity value 33.

We intuitively define the matching confidence γ_p for a pixel to be how distinctive the global minimum in the matching cost profile is:

$$\gamma_p = \begin{cases} 1 - \frac{c_p^{1st}}{c_p^{2nd}} & c_p^{2nd} > T_c \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

where c_p^{1st} and c_p^{2nd} are the best (lowest) and the second best matching cost of p, respectively. T_c is a small threshold value (set to $1e-3$ in our implementation) to avoid division by zero. $\gamma_p \in [0, 1]$ can be used to measure ambiguous matching caused by poor SNR. We experimentally found the confidence value itself is insufficient to model complex matching errors especially those caused by occlusions. To that end, we apply the left-right consistency check [7] to disambiguate occlusion pixels. For each pixel $p \in I$, if p violates the equality $D(p) = D'(p - D(p))$ then p is declared as occluded and added to the occlusion set O. We also observed that pixels with large disparity variations in a small neighborhood are prone to erroneously matched. Thus, if pixel p fails to satisfy the following inequality

$$|D(p) - \frac{\sum_{q \in \Phi_p} D(q)}{|\Phi_p|}| \leq 1, \tag{5}$$

it is treated as suspicious and added to the questionable set Q. Here Φ is a 3×3 neighborhood centered at pixel p. Note that both O and Q are $\emptyset$ at the beginning. Finally we define a binary map, denoted by M, on the reference image I. $M(p)$ is set to 1 if $p \in O \cup Q$ and 0 otherwise.

We define the reliability measure ρ_p for a pixel p using the conditional probability that p's estimated disparity is correct given γ_p and $M(p)$, as

$$\rho_p \propto P(\beta_p = 1|\gamma_p, M(p)). \tag{6}$$

By further assuming γ_p and $M(p)$ are independent variables we arrive at

$$\rho_p \propto P(\beta_p = 1|\gamma_p)P(\beta_p = 1|M(p)). \tag{7}$$

Likewise, p's unreliability measure $\overline{\rho_p}$ can be derived from equations 6 and 7 as,

$$\begin{aligned} \overline{\rho_p} &\propto P(\beta_p = 0|\gamma_p, M(p)) = P(\beta_p = 0|\gamma_p)P(\beta_p = 0|M(p)) \\ &= (1 - P(\beta_p = 1|\gamma_p))(1 - P(\beta_p = 1|M(p))). \end{aligned} \tag{8}$$

Our cost function U is constructed from equations 7 and 8 using the negative log-likelihood as

$$U_p(\beta_p) = \begin{cases} -\ln(\kappa_p \cdot \rho_p) & \beta_p = 1 \\ \\ -\ln(\kappa_p \cdot \overline{\rho_p}) & \beta_p = 0, \end{cases} \tag{9}$$

where κ_p in equation 9 is a normalization constant such that $\kappa_p(\rho_p + \overline{\rho_p}) = 1$.

Instead of using hard-coded $P(\beta_p = 1|\gamma_p)$ and $P(\beta_p = 1|M(p))$ we estimate the corresponding conditional probabilities based on stereo data sets with ground-truth disparities provided by [21,22]. In our experiment a total number of 13 stereo data sets are used to learn the parameters. We carefully select the training set so the matching difficulties of these stereo images span a wide range.

Since $M(p) \in \{0, 1\}$, $P(\beta_p = 1|M(p))$ can be easily estimated. After computing winner-take-all disparity maps $\{D, D'\}$ and the mask M, the conditional probability mass function is obtained by comparing masked and unmasked disparities with their ground-truth values and recording the correct rate. Our learned conditional probability mass function for $P(\beta_p = 1|M(p))$ is given in by:

$$P(\beta_p = 1|M(p)) = \begin{cases} 0.19 & M(p) = 1 \\ \\ 0.58 & M(p) = 0. \end{cases} \tag{10}$$

Learning $P(\beta_p = 1|\gamma_p)$ needs slightly more effort. Since $\gamma_p \in [0, 1]$, we discretize the continuous interval $[0, 1]$ into 200 bins. Pixel p is thus assigned to $bin_{\tilde{p}}$, whose corresponding value $val_{\tilde{p}}$ is most close to γ_p. In $bin_{\tilde{p}}$ the percentage of pixels that have correct estimated disparities is an approximation to $P(\beta_p = 1|val_{\tilde{p}})$ statistically. In Fig. 3 we plot the sampled conditional probability in red. There is a heavy tail in the plot because much fewer pixels are found with high matching confidence. Another interesting observation is that the probability actually

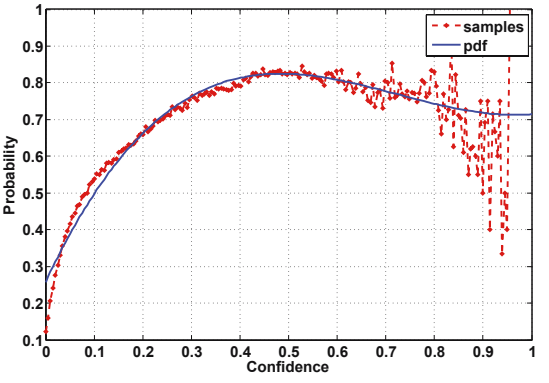


Fig. 3. The sampled conditional probabilities (red) and the cubic function computed via least square fitting to approximate the continuous pdf are given (blue)

starts decreasing when the confidence exceeds a certain value around 0.5. This phenomenon implies high confidences may partially result from occlusion in the image. To obtain a continuous probability density function (pdf) we apply a curve fitting using the sampled probability values given by bins that contain enough number of pixels. In this paper, bins whose number of pixels are less than 100 are discarded. Through the least square fitting we found the cubic function in 11 experimentally approximates the pdf quite well. The plot of this function is shown in Fig. 3 in blue.

$$P(\beta = 1|x) \approx 2.02x^3 - 4.38x^2 + 2.82x + 0.257 \tag{11}$$

Smoothness Term. The second term in equation 3 encourages spatial coherence and is helpful to remove matching outliers caused by noise and occluded pixels that survived the left-right check. The binary function V is defined as

$$V(\beta_p, \beta_q) = \lambda|\beta_p - \beta_q|. \tag{12}$$

The parameter λ controls the strength of the smoothness and is set to 0.5 throughout our experiments. Note that it is the only parameter designed experimentally in equation 3.

By treating the data term as an evidence function and the smoothness term as a compatibility function, equation 3 is the standard Gibbs function defined on a Markov network. As a two-label problem, the global optimal configuration which minimizes equation 3 can be exactly solved by GC. Although BP cannot guarantee the global optimal solution, our experiments show BP works equally successful on this problem in practice. In Fig. 4 we demonstrate the winner-take-all disparity map and the computed $GCPs$ of the Teddy data set using BP in the first two columns. The outliers in the candidate $GCPs$ are labeled in red. The matching density is 38% and the corresponding outlier percentage is 0.23%. It can be seen $GCPs$ from our approach are almost free of outliers.

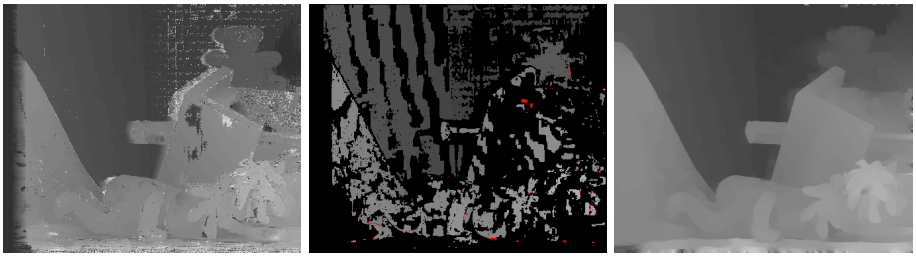


Fig. 4. From left to right: disparity map from winner-take-all method with cost aggregation; computed $GCPs$ of the Teddy data set and the outliers in the candidate $GCPs$ are labeled in red; $\overline{D}$ of the Teddy data set computed from the candidate $GCPs$ using adaptive propagation

2.3 Adaptive Propagation

The motivation of adaptive propagation is to allow propagation into regions where local matching costs are treated as unreliable by the proposed stable matching extraction algorithm. Our adaptive propagation algorithm is given as input the reference image I together with a semi-dense disparity map D_S on which the unreliable disparities D_S^U are left to be determined by those reliable disparities D_S^R. The objective is to infer D_S^U based on D_S^R and output a dense disparity map $\overline{D}$.

We wish to minimize the difference between the disparity of pixel p and the weighted average of the disparities at p's neighbors. A target cost function can be defined as

$$E(\overline{D}) = \sum_{p \in I} (\overline{D}(p) - \sum_{q \in N_p} \alpha_{pq} \overline{D}(q))^2, \tag{13}$$

where N_p is p's second order neighborhood system and α_{pq} is a weighting function that satisfies

$$\sum_{q \in N_p} \alpha_{pq} = 1. \tag{14}$$

The adaptation is based on the computation of the weighting function α. α_{pq} should be large when p and q have similar intensities and small otherwise. α controls the propagation strength so propagation stops near intensity edges, which are very likely to be depth discontinuities. In our experiments we adopt the weighting function defined in [8] as

$$\alpha_{pq} \propto 1 + \frac{(I(p) - \overline{I}_p)(I(q) - \overline{I}_p)}{\sigma_p^2}, \tag{15}$$

where $\overline{I}_p$ and σ_p^2 are the mean and variance of the intensities in a 5×5 window around p. By taking those stabled matched disparities D_S^R as constraints we can compute $\overline{D}$ by solving a system of sparse linear equations from a number of existing techniques [23]. When handling large stereo images, the resulting linear

system cannot be efficiently solved due to the large problem size. We therefore downsample both I and D_S to a reasonable size and upsample the solution to get $\overline{D}$. For example given a 1200×1000 image a downsampling factor 4 is used in our experiments. The corresponding $\overline{D}$ of the Teddy data set is shown in the third column of Fig. 4.

The cost function defined by equation 13 is not new. Levin et al. [8] use the same cost function to propagate color for colorization purpose. [24] minimizes similar function in segmentation. But based on our knowledge we believe it has not been used for stereo matching before. Different from tasks like colorization that needs only visually plausible result, stereo requires high accuracy estimate while such disparity propagation strategy clearly cannot guarantee enough accuracy. However, since our objective is not to locate the true disparity, this simple approach suits our problem pretty well by offering us a rough idea where the correct disparity is.

2.4 Search Range Selection

Given the disparity map D from winner-take-all together with the disparity map $\overline{D}$ from adaptive propagation our final per-pixel disparity search range is designed as follows: For each pixel p, first we compute a uncertainty radius Υ_p as

$$\Upsilon_p = \max(\frac{|D(p) - \overline{D}(p)|}{2}, 1). \tag{16}$$

Υ_p measures how well the disparity estimated from matching costs agrees with the disparity acquired from stably matched pixels. If Υ_p is large, we intuitively think the uncertainty is high and assign a large search range to p. Otherwise, we limit the search range for p since its matching cost is quite stable. The final disparity candidate set S_p is set to

$$\begin{aligned} S_p = &\{\max(d^{min}, D(p) - \Upsilon_p), \ldots, \min(D(p) + \Upsilon_p, d^{max})\} \bigcup \\ &\{\max(d^{min}, \overline{D}(p) - \Upsilon_p), \ldots, \min(\overline{D}(p) + \Upsilon_p, d^{max})\}. \end{aligned} \tag{17}$$

3 Experimental Results

In this section we report the results of applying our method on various stereo images and comparing against some existing algorithms that are related to our work. The side width of the window for adaptive cost aggregation is set to 33. Two constant aggregation parameters γ_c and γ_g defined in [18,19] are set to 12 and 40 respectively throughout.

We first evaluate the effectiveness of our stable matching extraction algorithm using ground-truth disparities and compare it with several semi-dense stereo algorithms. Note that there is no intersection between the training set and data sets we used to evaluate our algorithm. Table 1 shows the comparisons of different approaches including the current state-of-the-art [11]. Density D is the

Table 1. Performance comparison of the proposed stable matching detection method with other semi-dense stereo approaches

Algorithm	Tsukuba		Sawtooth		Venus		Map	
	$D(\%)$	e(%)	$D(\%)$	e(%)	$D(\%)$	e(%)	$D(\%)$	e(%)
Ours	**72**	**0.22**	**66**	**0.07**	**53**	**0.08**	**37**	**0.04**
RDP [11]	76	0.32	89	0.07	73	0.18	86	0.07
Veksler [13]	75	0.36	87	0.54	73	0.16	87	0.01
Sara [12]	45	1.40	52	1.60	40	0.80	74	0.30

Table 2. The performance of the proposed search space reduction algorithm. $\Re$ is the percentage of the reduced problem size. $\hbar$ is the percentage of pixels whose correct disparity values are preserved by our reduced search space.

	Tsukuba	Venus	Teddy	Cones
$\Re(\%)$	70.6	75.1	80.3	83.4
$\hbar(\%)$	99.1	99.7	97.3	97.5

percentage of reliable matching detected by the algorithm and error e is the percentage of wrong estimations (error $> \pm 1$ true disparity in non-occluded areas) in the semi-dense disparity maps. Numbers other than ours are from [11]. Note that all other three algorithms are designed for solving the stereo matching problem. Ours, however, only decides whether the disparity given by the best cost value is correct or not. This evaluation shows that our approach produces the most error-free disparity maps (in all but one case) and can preserve reasonable densities.

Table 2 shows our search space reduction rates. The stereo images we used for the test are the four standard data sets provided by the Middlebury stereo evaluation system [1]. Besides the reduction rate $\Re$ defined in equation (2), we further define the hit rate $\hbar$ as the percentage of pixels whose correct disparity values (within ± 1 true disparity) are preserved by our reduced search space. As can be seen, our algorithm is able to achieve 77% reduction rate while preserve 98.5% correct disparities in the reduced search space on average. The results are evaluated on the whole image domain including occlusion areas.

We also modify the traditional BP implementation such that each node can take different number of labels. By doing so the message length is no longer globally defined and memory requirement is greatly reduced. We use our BP implementation to compare the reconstruction quality given by full search range and the reduced search space generated from our framework. The evaluation results from [1] are demonstrated in Table 3. The second row shows the error percentages in non-occlusion areas using the full search space and the third row provides the corresponding error rates from our reduced space. As shown although the quality is not as good as the full search algorithm the differences are not significant. Considering the size of our search space is only 20% to 30% of the full search range these numbers are actually quite satisfactory. Some disparity maps are given in Fig. 5 for comparison.

Table 3. The comparison of reconstruction quality by using full search space (second row) and our reduced search space (third row). Disparity maps generated from BP are evaluated using Middlebury benchmark.

	Tsukuba	Venus	Teddy	Cones
Full range e(%)	1.06	0.78	7.59	5.26
Reduced space e(%)	1.10	1.02	7.96	6.49

Fig. 5. The first row shows results from full range algorithm and the second row demonstrates results from our reduced search space

At last we test our algorithm using high resolution stereo images. The image resolution is 1200×1000 and the full disparity search range is 120 pixels. Traditional Belief Propagation will require more than $2GB$ memory to only store the floating point messages. Our proposed algorithm, however, is able to reduce over 80% disparity search space so the remaining problem is solvable by ordinary computers[1].

4 Conclusions

In this paper we propose a novel algorithm for reducing search space for MRF-based stereo. The algorithm is simple yet very effective and can be used as a preprocessor for both BP and GC-based solvers. Interesting future work includes: applying our algorithm to facilitate high-resolution, large scale stereo reconstruction; investigating the performance gain of Belief Propagation and Graph Cuts given the reduced search spaces.

[1] The high resolution stereo images and the resulting disparity maps are available at www.vis.uky.edu/∼wangl

Acknowledgements

This work is supported in part by a US National Science Foundation grant IIS-0448185 and a grant from the US Department of Homeland Security.

References

1. Scharstein, D., Szeliski, R.: `vision.middlebury.edu/stereo/`
2. Sun, J., Zheng, N.N., Shum, H.Y.: Stereo matching using belief propagation. IEEE Trans. on Pattern Analysis and Machine Intelligence 25(7), 787–800 (2003)
3. Boykov, Y., Veksler, O., Zabih, R.: Markov random fields with efficient approximations. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 648–657 (1998)
4. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. on Pattern Analysis and Machine Intelligence 23(11), 1222–1239 (2001)
5. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. on Pattern Analysis and Machine Intelligence 26(9), 1124–1137 (2004)
6. Bobick, A.F., Intille, S.S.: Large occlusion stereo. Int. J. of Computer Vision, 181–200 (November 1999)
7. Egnal, G., Wildes, R.P.: Detecting binocular half-occlusions: empirical comparisons of five approaches. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(8), 1127–1133 (2002)
8. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. In: Proc. of ACM SIGGRAPH, pp. 689–694 (August 2004)
9. Veksler, O.: Reducing search space for stereo correspondence with graph cuts. In: British Machine Vision Conf., vol. 2, pp. 709–718 (2006)
10. Yu, T., Lin, R.S., Super, B., Tang, B.: Efficient message representations for belief propagation. In: Proc. of Intl. Conf. on Computer Vision (2007)
11. Gong, M., Yang, Y.H.: Fast unambiguous stereo matching using reliability-based dynamic programming. IEEE Trans. on Pattern Analysis and Machine Intelligence 27(6), 998–1003 (2005)
12. Sara, R.: Finding the largest unambiguous component of stereo matching. In: Proc. of Europ. Conf. on Computer Vision, pp. 900–914 (2002)
13. Veksler, O.: Extracting dense features for visual correspondence with graph cuts. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 689–694 (2003)
14. Cech, J., Sara, R.: Efficient sampling of disparity space for fast and accurate matching. In: Intl. Workshop on Benchmarking Automated Calibration, Orientation, and Surface Reconstruction from Images (2007)
15. Zhang, Z., Shan, Y.: A progressive scheme for stereo matching. In: Europ. Workshop on 3D Structure from Multiple Images of Large-Scale Environments (July 2000)
16. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M.F., Rother, C.: A comparative study of energy minimization methods for markov random fields. In: Proc. of Europ. Conf. on Computer Vision, vol. 2, pp. 19–26 (2006)

17. Birchfield, S., Tomasi, C.: A pixel dissimilarity measure that is insensitive to image sampling. IEEE Trans. on Pattern Analysis and Machine Intelligence 20(4), 401–406 (1998)
18. Wang, L., Liao, M., Gong, M., Yang, R.: High-quality real-time stereo using adaptive cost aggregation and dynamic programming. In: Intl. Symposium on 3D Data Processing, Visualization and Transmission, pp. 798–805 (2006)
19. Yoon, K.J., Kweon, I.S.: Locally adaptive support-weight approach for visual correspondence search. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 924–931 (2005)
20. Yang, Q., Wang, L., Yang, R., Stewenius, H., Nister, D.: Stereo matching with color-weighted correlation, hierachical belief propagation and occlusion handling. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 2347–2354 (2006)
21. Hirschmuller, H., Scharstein, D.: Evaluation of cost functions for stereo matching. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (2007)
22. Scharstein, D., Pal, C.: Learning conditional random fields for stereo. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (2007)
23. Press, W., Flannery, B., Teukolsky, S., Vetterling, W.: Numerical recipes inc. (1988)
24. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)

Estimating 3D Face Model and Facial Deformation from a Single Image Based on Expression Manifold Optimization

Shu-Fan Wang and Shang-Hong Lai

Department of Computer Science,
National Tsing Hua University, Taiwan
{shufan,lai}@cs.nthu.edu.tw

Abstract. Facial expression modeling is central to facial expression recognition and expression synthesis for facial animation. Previous works reported that modeling the facial expression with low-dimensional manifold is more appropriate than using a linear subspace. In this paper, we propose a manifold-based 3D face reconstruction approach to estimating the 3D face model and the associated expression deformation from a single face image. In the training phase, we build a nonlinear 3D expression manifold from a large set of 3D facial expression models to represent the facial shape deformations due to facial expressions. Then a Gaussian mixture model in this manifold is learned to represent the distribution of expression deformation. By combining the merits of morphable neutral face model and the low-dimensional expression manifold, we propose a new algorithm to reconstruct the 3D face geometry as well as the 3D shape deformation from a single face image with expression in an energy minimization framework. Experimental results on CMU-PIE image database and FG-Net video database are shown to validate the effectiveness and accuracy of the proposed algorithm.

Keywords: 3D Modeling, Facial Expression, Manifold.

1 Introduction

3D human face modeling from images is a very popular topic with many applications, such as facial animation, face recognition, model-based facial video communication, etc. Previous works on 3D head modeling from a single face image utilized prior information on 3D head models. However, it is difficult to accurately reconstruct the 3D face model from a single face image with expression since the facial expression induces 3D face model deformation in a complex manner. The main challenge is the coupling of the neutral 3D face model and the 3D deformation due to expression, thus making the 3D model estimation from a single face image with expression very difficult. In this paper, we propose a 3D face model reconstruction algorithm that can recover the 3D face model as well as the 3D expressional deformation from a single face image with expression. Our algorithm integrates the linear and non-linear subspace representations for a prior 3D neutral morphable model and the probabilistic manifold-based 3D expressional deformation.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 589–602, 2008.

1.1 Related Work

Model-based statistical techniques have been widely used for robust human face modeling. Most of the previous 3D face reconstruction techniques require more than one face image to achieve satisfactory 3D human face modeling. Another approach for 3D face reconstruction from a single image is to simplify the problem by using a statistical head model as the prior. For example, Blanz and Vetter [1] proposed an algorithm for 3D face model reconstruction by minimizing the discrepancies between the face image and the corresponding image rendered from a morphable 3D head model under a suitable illumination condition. Later, this technique was successfully applied to near-neutral face recognition [2] and achieved a high recognition rate.

In practice, it is difficult to estimate the lighting condition from a single face image. In addition, the facial expression variation could be the most critical issue in the 3D face model reconstruction process. For 3D facial expression analysis, most of the previous works [3,4,5,6] focused on avoiding or reducing the effect of facial expression to achieve accurate 3D face recognition. There are also several methods proposed to analyze facial expression for different applications with the aid of a 3D face model [7,8] or multiple views [9]. Blanz et al. [10] further extended their previous works [1,2] to handle facial expression by collecting 35 expression 3D scans from an individual and reanimated the faces in images by linear-subspace analysis. However, they ignored the variations across individuals and the styles of different individuals can not be well represented. The expression and light condition re-targeting technique proposed in [11] can transfer the expression and illumination from a 3D scan to a reconstructed neutral 3D face model by using the estimated spherical harmonic light field and existing facial expression motion field. Recently, non-linear embedding methods such as ISOMAP [12], locally linear embedding [13], and global coordinate of local linear method [14] were proposed to handle high-dimensional non-linear data, which could be more appropriate to model the facial expression variations. Continuous facial expression deformations in images represent a smooth manifold in the high-dimensional space and the intrinsic dimension could be much smaller. Researchers have also developed manifold-based learning methods for facial expressions and applied them to 2D face expression recognition or synthesis [15,16,17]. Later, a manifold-based method [18] was proposed to register 3D scans and transfer expression from one 3D face to another by using the unified manifold space analysis.

1.2 Contribution

In this paper, we propose an algorithm to reconstruct a 3D face model with the associated expressional deformation directly from a single 2D face image with expression based on linear and non-linear subspace analysis. We propose a novel approach which combines probabilistic manifold embedding and prior 3D eigenhead model to perform this task. The manifold is not constructed to represent the whole expressional 3D models, but only model the deformation from neutral

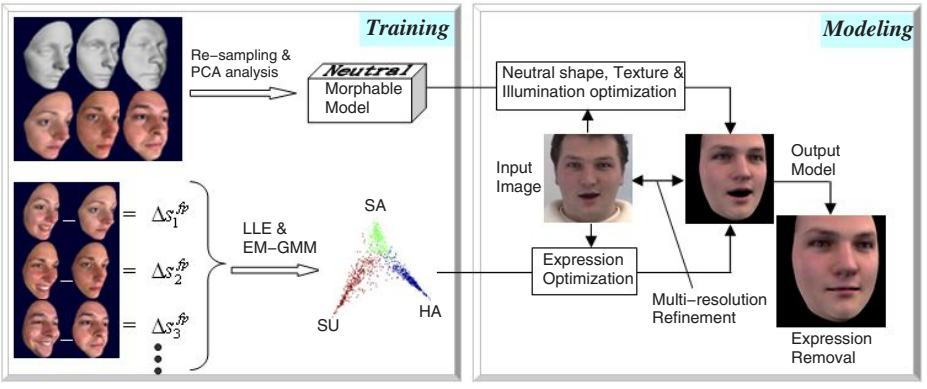


Fig. 1. The framework of our 3D expression reconstruction

3D face to the corresponding expressional 3D face. The parameters estimated during the reconstruction process are sampled from the probability distribution on the neutral eigenface space as well as the low-dimensional expression manifold space and they minimize the total-image re-projection error simultaneously.

In summary, the contributions of this proposed approach are listed as follows: First, we propose to reconstruct the complete 3D face model with expression deformation from a single image without existing action unit, motion field, or any expression style assumption. Second, a probabilistic nonlinear 3D expression manifold is learned from a large set, more than 1000, of 3D facial expression models to represent the general deformations from facial expressions. The manifold reduces the complexity of the reconstruction problem and therefore makes it easier and more robust to track the continuous expression deformation in image sequence. Third, from the learned expression manifold, the reconstructed 3D face model can be re-targeted to any expression content, style or even mixture of them in a more general way.

The global view of our training and reconstruction process is shown in Fig.1. Inter and intra face deformation modeling will be introduced in the following two sections. In section 2, we briefly review the technique of morphable model and spherical harmonics for shape and illumination approximation, respectively. In section 3, we describe how to estimate the probabilistic manifold for facial expression deformations. The main steps of the reconstruction process and parameter estimation are described in section 4. We demonstrate the effectiveness and accuracy of the proposed algorithm through experiments on simulated and real images in section 5. The final section concludes this paper.

2 Morphable Model and Spherical Harmonics Illumination

In this section, we will first briefly describe the morphable model used for 3D face model reconstruction from a single face image. On the other hand, spherical

harmonic bases [19] have been used to approximate the face images of a 3D face model under different illumination conditions. The morphable model and spherical harmonic bases are employed in our algorithm to approximate a 3D face model with a small number of parameters and the corresponding face images under different lighting conditions, respectively.

2.1 Morphable Model

Morphable model provides the prior knowledge of neutral 3D face geometry as well as the texture. The geometry of a face model can be represented as $S = (x_1, y_1, z_1 \cdots, x_N, y_N, z_N) \in \Re^{3N}$ and the appearance can be represented as a vector $T = (r_1, g_1, b_1 \cdots, r_N, g_N, b_N) \in \Re^{3N}$, where N is the total number of vertices in a 3D model. Therefore, the geometry and texture of a 3D face model can be approximated by a mean and a linear combination of several eigenhead basis vectors:

$$\begin{aligned} S(\boldsymbol{\alpha}) &= \bar{S} + \sum_{i=1}^{m} \alpha_i \mathbf{s}_i \\ T(\boldsymbol{\beta}) &= \bar{T} + \sum_{i=1}^{m} \beta_i \mathbf{t}_i \end{aligned} \tag{1}$$

where $\bar{S}$ and $\bar{T}$ are the mean shape and texture vector, s_i and t_i are the i-th eigen-shape basis and the i-th eigen-texture basis of the morphable model, respectively, and $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_m]$ and $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_m]$ contain the shape and texture coefficients, respectively, to represent the 3D head model. In this work, we use the 3D face scans and images from BU-3DFE database [20] as the training faces for generating the morphable model. In order to obtain the point-wise correspondence of all the 3D faces, we build a generic model shown in Fig. 2(left). The 3D scans and images from the neutral faces in BU-3DFE are registered to the generic model and re-sampled to compute the shape and texture bases in the morphable model. An example of the training data is depicted in Fig. 2.

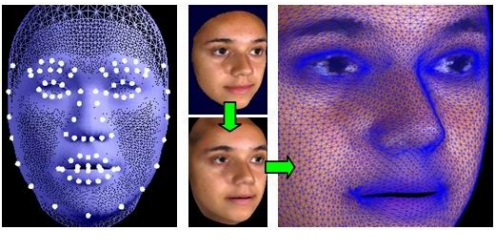


Fig. 2. The preprocessing of training data. Left: The *generic face model* labeled with 83 feature points. Center: The upper row is the *original face scan* and the lower is the model after registration, re-sampling and smoothing. Right: The *triangulation detail* of the processed model.

2.2 Spherical Harmonic Bases

Spherical harmonic bases [19] have been used to approximate the images of a 3D model under a wide variety of lighting conditions. These bases could be determined by the surface normal $\mathbf{n}$ and the albedo $\boldsymbol{\lambda}$. In this case, the albedo $\boldsymbol{\lambda}$ can be approximated by the texture part of the morphable model $T(\boldsymbol{\beta})$ in Eq. (1). It is easier to describe the basis if we write it as a function of x, y, z instead of the angle of spherical coordinates. Let n_x, n_y, n_z denote the unit normal vector at the corresponding 3D point, denote the vector containing the albedos of the object and the operator .* denote the element-wise product. With the notation used in [21] , the first nine spherical harmonic bases for the image intensity at a 3D point are given bellow:

$$\begin{aligned}
&b_{00} = \tfrac{\lambda}{\sqrt{4\pi}}, b^e_{11} = \sqrt{\tfrac{3}{4\pi}}\lambda. * n_x, \\
&b^o_{11} = \sqrt{\tfrac{3}{4\pi}}\lambda. * n_y, b^e_{10} = \sqrt{\tfrac{3}{4\pi}}\lambda. * n_z, \\
&b^o_{21} = 3\sqrt{\tfrac{5}{12\pi}}\lambda. * n_y n_z, b^o_{22} = 3\sqrt{\tfrac{5}{12\pi}}\lambda. * n_y n_z, \\
&b^e_{21} = 3\sqrt{\tfrac{5}{12\pi}}\lambda. * n_x n_z, b^e_{22} = \tfrac{3}{2}\sqrt{\tfrac{5}{12\pi}}\lambda. * (n_x n_x - n_y n_y) \\
&b_{20} = \tfrac{1}{2}\sqrt{\tfrac{3}{4\pi}}\lambda. * (2n_z n_z - n_x n_x - n_y n_y)
\end{aligned} \tag{2}$$

By concatenating the nine spherical harmonic bases for all the N 3D points into the matrix $\mathbf{B}(\lambda, \mathbf{n}) \in \Re^{N\times 9}$, we can approximate the image of a 3D model under arbitrary illumination conditions with a linear combination of the bases as follows:

$$I_{\text{model}} = \mathbf{B}\ell \tag{3}$$

where $\ell \in \Re^9$ is a 9-dimensional weighting vector.

3 Probabilistic Manifold Embedding for the Deformations from Facial Expressions

There are several techniques proposed recently to infer the intrinsic structure of the non-linear data. After some experiments, our results show that locally linear embedding (LLE) and Laplacian eigenmaps are suitable for modeling the expression deformations. For our training data set, LLE outperforms the Laplacian eigenmaps. In this section, we present how to build a probabilistic model in a non-linear low-dimensional manifold for representing of the 3D deformation due to facial expression.

3.1 Low-Dimensional Embedding

We employ the LLE [13] to achieve a low-dimensional non-linear embedding of M expression deformations $\Delta\mathbf{s}_i^{fp}$ obtained by registering between the 3D models of a subject with and without expression in the BU-3DFE database [20]:

$$\Delta\mathbf{s}_i^{fp} = \mathbf{S}_{Ei}^{fp} - \mathbf{S}_{Ni}^{fp} \tag{4}$$

where $\mathbf{S}_{Ei}^{fp} = \{x_1^E, y_1^E, z_1^E, \cdots x_n^E, y_n^E, z_n^E\} \in \Re^{3n}$ is a set of feature points representing the i-th 3D face geometry with facial expression, and similarly, $\mathbf{S}_{Ni}^{fp}$ denotes the set of feature points of the corresponding 3D neutral face. In our experiment, we used 83 feature landmarks for each face scan obtained from the database BU-3DFE, as shown in Fig. 2(a). The collection of M 3D expression deformations, denoted by $\Delta\mathbf{s}_i^{fp}$ for $i = 1, \cdots, M$, including the happy, sad and surprise expressions, are embedded to M 2D points on the manifold space. Fig. 3(a) shows the embedded two-dimensional manifold space.

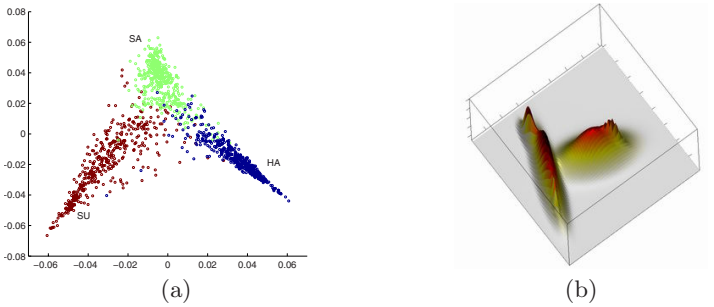
3.2 Probability Distribution in the Expression Manifold

As described in the previous sub-section, M training data sets are projected onto a 2D manifold, including different magnitude, content, and styles of expressions. Each expression of a certain individual forms a trajectory in the manifold 2D space. In order to represent the distribution of expression deformations, Gaussian Mixture Model (GMM) is used to approximate the probability distribution of the 3D expression deformation in this low-dimensional expression manifold as follows:

$$p_{GMM}(\mathbf{s}^{LLE}) = \sum_{c=1}^{C} \omega_c N(\mathbf{s}^{LLE}; \mu_c, \Sigma_c) \tag{5}$$

where ω_c is the probability of being in cluster c, $0 < \omega_c < 1$ and $\sum_{c=1}^{C} \omega_c = 1$, μ_c and Σ_c are the mean and covariance matrix for the c-th Gaussian distribution.

The expectation maximization (EM) algorithm is employed to compute the maximum likelihood estimation of the model parameters from the training data. We apply the EM-based Gaussian mixtures estimation [22] to build the unsupervised GMM in the LLE manifold space for 3D expression deformations. The colorful dots shown in Fig. 3(a) indicate the distribution of the manifold and the probability distribution of the estimated GMM in the expression manifold is shown in Fig. 3(b).



(a) (b)

Fig. 3. Low-dimensional manifold representation of expression deformations. (a) 3D expression deformations are projected onto the 2D *expression manifold*. (b) The *probability distribution* of the estimated GMM.

4 3D Model Reconstruction from a Single Face Image with Unknown Expression

In this section, we propose a new algorithm to reconstruct the 3D face model from a single image based on the learned neutral 3D face morphable model and the probabilistic 2D expression manifold model. We apply an iterative scheme to optimize the intra and inter deformation of 3D human face models. To be more robust during the optimization procedure, we first analyze the magnitude of the expression deformation. We quantify the deformation of every vertex in the original 3D space to measure the deformation magnitude. The color distribution shown in Fig. 4 denotes the magnitude distribution of the corresponding expression deformation and the unified magnitude vector is obtained by calculating the combination of the magnitudes from different expressions. From the above statistics on the deformation magnitudes for different expressions at all locations in the 3D face model, we can determine the weighting of each node in the face 3D model for the morphable model (neutral face) as well as the expression model. Therefore, the weighting of each 3D vertex j for a neutral face model, denoted by w_j^N, can be defined as:

$$w_j^N = \frac{mag_{\max} - mag_j}{mag_{\max} - mag_{\min}} \tag{6}$$

where $mag_{\max}$, $mag_{\min}$, and mag_j denote maximal, minimal and the j-th vertex's deformation magnitudes, respectively. Similarly, for facial expression modeling, the weighting of each 3D vertex j, denoted by w_j^E is defined by:

$$w_j^E = 1 - w_j^N \tag{7}$$

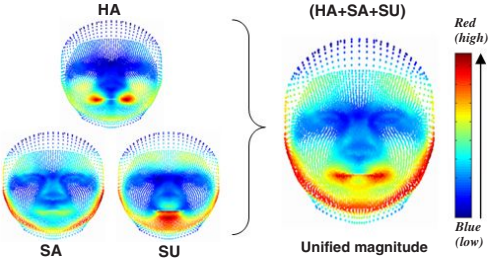


Fig. 4. The *magnitude distribution* of facial deformation under different expressions. The magnitudes of different expressions are drawn on the generic model.

4.1 Initialization

For the initialization of 3D model, we first estimate the shape parameters by minimizing the geometric distance of the landmark features. The minimization problem is given by:

$$\min_{f,\mathbf{R},\mathbf{t},\boldsymbol{\alpha}} \sum_{j=1}^{n} w_j^N \left\| \mathbf{u}_j - (\mathbf{P} f \mathbf{R} \hat{\mathbf{x}}_j(\boldsymbol{\alpha}) + \mathbf{t}) \right\| \tag{8}$$

where $\mathbf{u}_j$ denotes the coordinate of the j-th feature point in 2D image, $\mathbf{P}$ is an orthographic projection matrix, f is the scaling factor, $\mathbf{R}$ denotes the 3D rotation matrix, $\mathbf{t}$ is the translation vector, and $\hat{x}_j(\boldsymbol{\alpha})$ denotes the j-th reconstructed 3D feature point that is determined by the shape parameter vector $\boldsymbol{\alpha}$ as follows

$$\hat{x}_j = \bar{x}_j + \sum_{l=1}^{m} \alpha_l \mathbf{s}_l^j \tag{9}$$

The function minimization problem given in Eq. (8) can be solved by using the Levenberg-Marquart optimization [23] to find the 3D face shape vector and the pose of the 3D face as the initial solution for the 3D face model. In this step, the 3D neutral face model is initialized and the effect of the deformation from facial expression can be alleviated by using the weighting w_j^N for the neutral face model.

Since the magnitude, content, and styles of expressions are all embedded into the low-dimensional expression manifold, the only parameters for facial expression are the coordinate of $\mathbf{s}^{LLE}$. The initial $\mathbf{s}^{LLE}$ is set to (0, 0.01), which is located at the common border of different expressions on the expression manifold.

4.2 Parameters Optimization

After the initialization step, all the parameters are iteratively optimized in two steps which will be described in details subsequently.

Texture and Illumination Update. To recover the texture and illumination, we need to estimate texture coefficient vector $\boldsymbol{\beta}$ and then determine the illumination bases $\mathbf{B}$ and the corresponding spherical harmonic (SH) coefficient vector ℓ. From Eq. (2), spherical harmonic bases $\mathbf{B}$ are determined by the surface normal $\mathbf{n}$ and texture intensity $T(\boldsymbol{\beta})$. Therefore, with the surface normal $\mathbf{n}$ determined from the current 3D face model, the texture coefficient vector $\boldsymbol{\beta}$ and the SH coefficient vector ℓ can be estimated by solving the non-linear optimization problem:

$$\min_{\beta,\ell} \|\mathbf{I}_{input} - \mathbf{B}(T(\beta), \mathbf{n})\ell\| \tag{10}$$

According to different reflection properties in the face feature area and skin area, we define these two areas for more accurate texture and illumination estimation. Since the feature area is less sensitive to lighting variations, texture coefficient vector $\boldsymbol{\beta}$ are estimated based on minimizing the intensity errors for the vertices in the *face feature area*. On the other hand, the SH coefficient vector ℓ is determined by minimizing the image intensity errors in the *skin area*.

3D Face Shape Update. In this step, the inter and intra face deformation is estimated from the photometric approximation with the estimated texture parameters obtained from the previous step. Since we have the statistical models of

facial geometry and deformation, the shape parameters $\boldsymbol{\alpha}$, the expression parameters $\hat{\mathbf{s}}^{LLE}$ and the pose vector $\boldsymbol{\rho} = \{f, \mathbf{R}, \mathbf{t}\}$ can be estimated by maximizing the posterior probability(MAP) for a given input image $\mathbf{I}_{\text{input}}$ and the texture parameter vector $\boldsymbol{\beta}$. Similar with [2], we neglect the correlation between these parameters and the posterior probability can be written as follows:

$$\begin{aligned} &p(\boldsymbol{\alpha}, \boldsymbol{\rho}, \hat{\mathbf{s}}^{LLE} | \mathbf{I}_{\text{input}}, \boldsymbol{\beta}) \propto p(\mathbf{I}_{\text{input}} | \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}, \hat{\mathbf{s}}^{LLE}) \cdot p(\boldsymbol{\alpha}, \rho, \hat{\mathbf{s}}^{LLE}) \\ &\approx \exp\left(\frac{-\left\|\mathbf{I}_{\text{input}} - \mathbf{I}_{\exp}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}, \hat{\mathbf{s}}^{LLE})\right\|^2}{2\sigma_I^2}\right) \cdot p(\boldsymbol{\alpha}) \cdot p(\boldsymbol{\rho}) \cdot p(\hat{\mathbf{s}}^{LLE}) \end{aligned} \tag{11}$$

with

$$\mathbf{I}_{\exp}(\boldsymbol{\alpha}, \boldsymbol{\beta}, f, \mathbf{R}, \mathbf{t}, \hat{\mathbf{s}}^{LLE}) = \mathbf{I}(f\mathbf{R}(S(\boldsymbol{\alpha}) + \psi(\hat{\mathbf{s}}^{LLE})) + \mathbf{t}) \tag{12}$$

where σ_I is the standard deviation of the image synthesis error, and $\psi(\hat{\mathbf{s}}^{LLE}) : \Re^e \rightarrow \Re^{3N}$ is a nonlinear mapping function that maps the estimated $\hat{\mathbf{s}}^{LLE}$ from the embedded space with dimension $e = 2$ to the original 3D deformation space with dimension $3N$. Although the embedded manifold is globally nonlinear, it is constructed based on a neighbor-preserving mapping. Therefore, we use the nonlinear mapping function of the following form:

$$\psi(\hat{\mathbf{s}}^{LLE}) = \sum_{k \in NB(\hat{\mathbf{s}}^{LLE})} w_k \Delta \mathbf{s}_k \tag{13}$$

where $NB(\hat{s}^{LLE})$ is the set of nearest neighbor training data points to $\hat{\mathbf{s}}^{LLE}$ on the expression manifold,$\Delta \mathbf{s}_k$, is the 3D deformation vector for the k-th facial expression data in the training data set, and the weight w_k is determined from the neighbors by the same method described in LLE [13].

Since the prior probability of $\hat{\mathbf{s}}^{LLE}$ in the expression manifold is given by the Gaussian mixture model $p_{GMM}(\hat{\mathbf{s}}^{LLE})$ and $\boldsymbol{\alpha}$ is estimated by PCA analysis, maximizing the log-likelihood of the posterior probability in Eq.(11) is equivalent to minimizing the following energy function.

$$\begin{aligned} &\max\left(\ln p(\boldsymbol{\alpha}, \rho, \hat{\mathbf{s}}^{LLE} | \mathbf{I}_{\text{input}}, \boldsymbol{\beta})\right) \approx \\ &\min\left(\frac{\left\|\mathbf{I}_{\text{input}} - \mathbf{I}_{\exp}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \rho, \hat{\mathbf{s}}^{LLE})\right\|^2}{2\sigma_I^2} + \sum_{i=1}^{m} \frac{\alpha_i^2}{2\lambda_i} - \ln p(\rho) - \ln p_{GMM}(\hat{\mathbf{s}}^{LLE})\right) \end{aligned} \tag{14}$$

where λ_i denotes the i-th eigenvalue estimated with PCA for neutral 3D faces. Note that $p(\rho)$ can be simply a constant or modeled with a reasonable prior probability density function to impose constraints on the pose parameters.

4.3 Algorithm Summary

To reduce the possibility of being trapped into a local minimum solution, the proposed 3D face model reconstruction algorithm follows a coarse-to-fine optimization strategy in two respects:

- We build two morphable models for low and high resolutions and an L-level Gaussian pyramid for each input face image I_{input}. The level of I_{input} will decrease and the resolution of the morphable model will increase along with the iteration.
- The number of eigenhead bases used for neutral face modeling will also be increased as the iteration increases.

Our algorithm is summarized as follows:

1. Build an L-level Gaussian pyramid for the input image I_{input}, and set $level = L$. Initially we apply the low-resolution morphable model and set the number of eigenhead bases $m = 5$.
2. Initialize the pose and shape coefficients by the feature landmarks. Set the initial $\mathbf{s}^{LLE} = (0, 0.01)$. (Section 4.1)
3. Optimize the texture coefficient vector $\boldsymbol{\beta}$ and update the SH coefficient vector by solving Eq. (10). (Section 4.2)
4. Update the pose parameters, neutral face shape model parameters $\boldsymbol{\alpha}$, and expression parameters $\mathbf{s}^{LLE}$ simultaneously by minimizing the cost function in Eq. (14). (Section 4.2)
5. Set $level = level - 1$, and $m = m + 2$. Apply the high-resolution morphable model when $level \leq \frac{L}{2}$.
6. Repeat step 3, 4 and 5 until $level = 0$.

5 Experimental Results

In this section, we validate the proposed method by conducting several experiments on different data sets. In the following experiments, we used the same definition of facial feature points in BU-3DFE database [20] both in the training phase and the initial step of the reconstruction process for comparison of different methods. First, the 3D reconstruction accuracy is evaluated through experiments on the face images simulated from known 3D face models. The second experiment is conducted on CMU-PIE data set [24]. We also apply the proposed algorithm to achieve 3D modeling and tracking on the FG-Net [25] database with different facial expressions.

5.1 3D Reconstruction Accuracy

In order to measure the accuracy of reconstructed 3D face model, we randomly generate simulated face models from three sets of expression scans. These 3D face models are generated with different degrees, types, and style of expressions, including happiness, sadness, and surprise, under arbitrary illumination conditions. The images are rendered with OpenGL, as depicted in Fig. 5, and used as the input images for the 3D reconstruction.

For more quantitative measurement, we calculate the average error from the 3D differences of all vertices and normalize the error by the size, $edge_{cube}$, of the bounding cube containing the 3D face model:

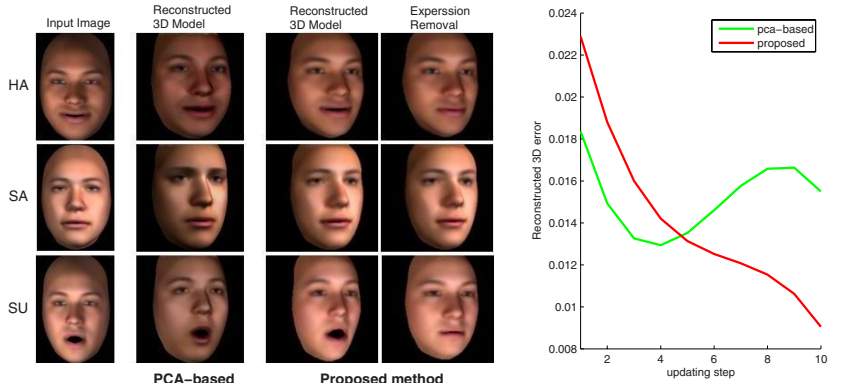


Fig. 5. 3D expression reconstruction from a single simulated face image. (a) The first column shows the *input images* and the second column are the results of PCA-based method. The third and the fourth columns represent the *reconstructed 3D face models* with expression, and those after *expression removal* by our proposed algorithm. (b) The curves are the average differences between reconstructed 3D face model and the ground truth along the updating steps with two different methods.

$$err = \frac{\sum_{j=1}^{N} \left\| \hat{v}_j - v_j^g \right\|}{N \cdot edge_{cube}} \tag{15}$$

where $\hat{v}_j$ is the j-th 3D vertex of the reconstructed face model, and v_j^g is the j-th 3D vertex of the ground truth. In this experiment, the proposed algorithm is also compared with the PCA-based method. For a fair comparison, the PCA model is built from the face models with neutral, happiness, sadness, and surprise expressions and the same optimization scheme is applied for the PCA method. The reconstructed expressional 3D faces and the models after expression removal (i.e. the neutral part $S(\boldsymbol{\alpha})$) are shown in Fig. 5. The average errors between the reconstructed 3D model and the ground truth are computed with Eq. (15) at every updating step. By optimizing the photometric approximation in the 2D image, the error of our 3D reconstruction is monotonically decreased as the updating steps increase. Comparatively, the results of the PCA method shown in Fig. 5 are not stable and the reconstructed 3D face models sometimes are not satisfactory. The unstable phenomenon of the PCA-based method in Fig. 5(b) may be caused by the unsuitable linear modeling of the manifold structured expression deformation, which leads to ambiguity in the optimization process.

5.2 Experiments on Real Images

We used the CMU-PIE data set [24], which contains face images under different poses, illumination and facial expressions, to test the proposed algorithm. The facial expressions in CMU-PIE include smile, blink and talk, and all the face

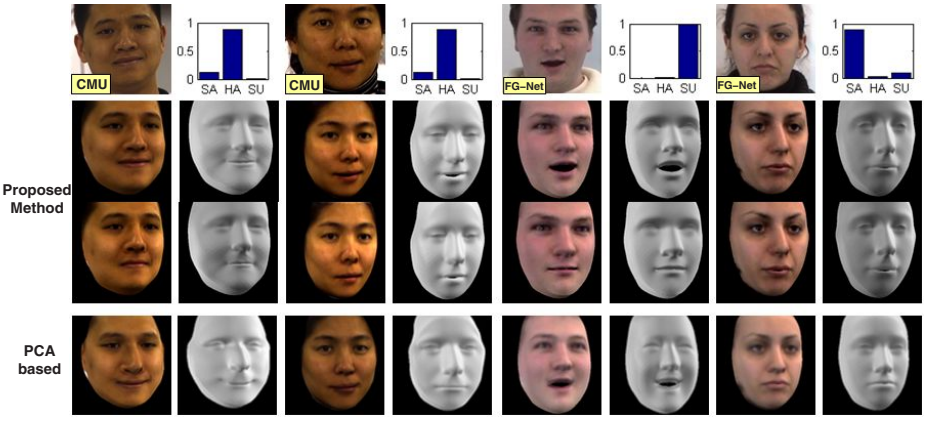


Fig. 6. 3D expression reconstruction from a single real image: the first row shows the *input images* and the bar graphs of the estimated *probabilities* for the expression modeling on the learned manifold. The second and third row represent our results, including the final *reconstructed expressive face models* and those after *expression removal*. The bottom row shows the results from the *PCA-based* method.

images with smile expressions are used in our experiments. For sadness and surprise expressions, we selected some frames of these two expressions in FG-Net database [25] as the input images in our experiments.

Most of the testing images got high probabilities for the corresponding expressions. Some of the reconstructed models and the corresponding probability distributions of different expressions are shown in Fig. 6. The bar graphs in Fig. 6 show the estimated probabilities for the expression modeling on the learned manifold, which also demonstrate the accuracy of facial expression estimation. The second and third rows show the reconstructed 3D face model with expressions under a novel view and the synthesized images after expression removal by using the proposed method. The bottom row shows the results of PCA-based method which can not provide satisfactory face model reconstruction. The attached supplemental materials contain more 3D face reconstruction results.

5.3 Experiments on Video Sequences

We also applied the proposed algorithm to estimate the 3D deformable face models from video sequences with facial expressions. The face videos with different facial expressions from the FG-Net database [25] are used as inputs to our algorithm. We labeled the feature landmarks in the first frame of each video sequence and reconstruct the 3D face model with the associated 3D expression deformation. In the subsequent frames, we update the parameters of pose, shape, and expression deformation by photometric approximation as described in Section 4.2. Some results of the testing sequences are shown in Fig. 7. Similarly, we include more experimental results in the attached supplemental materials.

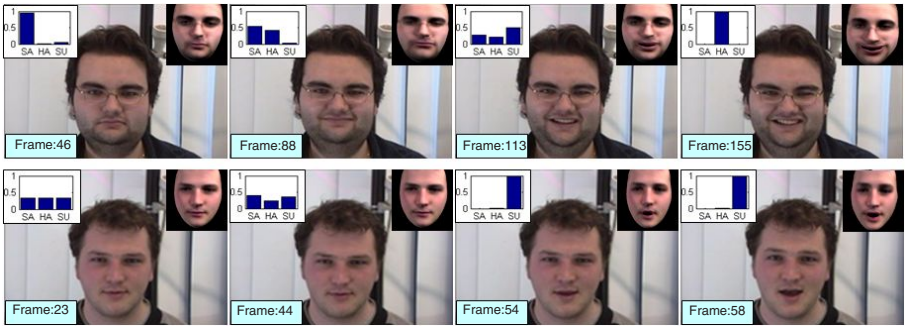


Fig. 7. 3D face reconstruction from video sequences in FG-Net video database

6 Conclusion

In this paper, we proposed a novel algorithm for reconstructing the 3D face model and the associated 3D expression deformation from a single expressional face image based on modeling the 3D expression deformation with the probabilistic manifold modeling and 3D neutral face models with the 3D morphable model. The non-linear expression deformation manifold and the linear morphable neutral face model are integrated in an optimization framework for this problem. With the combination of spherical harmonics, the texture and illumination condition can be measured more accurately, thus improving the 3D shape optimization. Based on this framework, we can include more different types of expression deformations into the training data to achieve 3D modeling or re-targeting with the learned probabilistic manifold for more general facial expression images. The experiments on the simulated and real images demonstrate the accuracy and effectiveness of the proposed 3D face reconstruction algorithm.

References

1. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d-faces. In: SIGGRAPH (1999)
2. Blanz, V., Vetter, T.: Face recognition based on fitting a 3d morphable model. PAMI 25(9), 1063–1074 (2003)
3. Bronstein, A., Bronstein, M., Kimmel, R.: Expression invariant 3d face recognition. AVBPA 2688, 62–70 (2003)
4. Bronstein, A., Bronstein, M., Kimmel, R.: Three dimensional face recognition. IJCV 64(1), 5–30 (2005)
5. Wang, Y., Pan, G., Wu, Z.: 3d face recognition in the presence of expression: a guidance-based constraint deformation approach. In: CVPR, pp. 1–7 (2007)
6. Kakadiaris, I.A., Passalis, G., Toderici, G., Murtuza, M.N., Lu, Y., Karampatziakis, N., Theoharis, T.: Three-dimensional face recognition in the presence of facial expressions: an annotated deformable model approach. PAMI (2007)
7. Wen, Z., Huang, T.: Capturing subtle facial motions in 3d face tracking. In: ICCV (2003)

8. Zalewski, L., Gong, S.: Synthesis and recognition of facial expressions in virtual 3d views. FGR (2004)
9. Pighin, F., Hecker, J., Lischinski, D., Szeliski, R., Salesin, D.H.: Synthesizing realistic facial expressions from photographs. In: SIGGRAPH, pp. 75–84 (1998)
10. Blanz, V., Basso, C., Poggio, T., Vetter, T.: Reanimating faces in images and video. EG (2003)
11. Zhang, L., Wang, Y., Wang, S., Samaras, D., Zhang, S., Huang, P.: Image-driven re-targeting and relighting of facial expressions. CGI (2005)
12. Tenenbaum, J., de Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science (2000)
13. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science (2000)
14. Roweis, S., Saul, L., Hinton, G.: Global coordination of local linear models. Neural Information Processing Systems 14 14, 889–896 (2001)
15. Chang, Y., Hu, C., Turk, M.: Manifold of facial expression. In: Proc. IEEE intern. Workshop AMFG (2003)
16. Chang, Y., Ho, C., Turk, M.: Probabilistic expression analysis on manifolds. In: CVPR (2004)
17. Hu, C., Chang, Y., Feris, R., Turk, M.: Manifold based analysis of facial expression. In: IEEE Workshop on Face Processing in Video (2004)
18. Wang, Y., Huang, X., Lee, C.S., Zhang, S., Li, Z.: High resolution acquisition, learning and transfer of dynamic 3-d facial expressions. EG (2004)
19. Basri, R., Jacobs, D.: Lambertian reflectance and linear subspaces. PAMI 25(2), 218–233 (2003)
20. Yin, L., Wei, X., Sun, Y., Wang, J., Rosato, M.: A 3d facial expression database for facial behavior research. FG, 211–216 (2006)
21. Zhang, L., Samaras, D.: Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics. PAMI 28(3), 351–363 (2006)
22. Bilmes, J., Gentle, A.: Tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. International Computer Science Institute (1998)
23. Levenberg, K.: A method for the solution of certain non-linear problems in least squares. Quarterly of Applied Mathematics 2(2), 164–168 (1944)
24. Sim, T., Baker, S., Bsat, M.: The cmu pose, illumination, and expression database. PAMI, 1615–1618 (2003)
25. FG-Net database, `http://www.mmk.ei.tum.de/waf/fgnet/feedtum.html`

3D Face Recognition by Local Shape Difference Boosting

Yueming Wang[1], Xiaoou Tang[1], Jianzhuang Liu[1], Gang Pan[2], and Rong Xiao[3]

[1] Dept. of Information Engineering, The Chinese University of Hong Kong
ymingwang@gmail.com, {jzliu,xtang}@ie.cuhk.edu.hk
[2] College of Compute Science, Zhejiang University
gpan@zju.edu.cn
[3] Microsoft Research Asia

Abstract. A new approach, called *Collective Shape Difference Classifier* (CSDC), is proposed to improve the accuracy and computational efficiency of 3D face recognition. The CSDC learns the most discriminative local areas from the *Pure Shape Difference Map* (PSDM) and trains them as weak classifiers for assembling a collective strong classifier using the real-boosting approach. The PSDM is established between two 3D face models aligned by a posture normalization procedure based on facial features. The model alignment is self-dependent, which avoids registering the probe face against every different gallery face during the recognition, so that a high computational speed is obtained. The experiments, carried out on the FRGC v2 and BU-3DFE databases, yield rank-1 recognition rates better than *98*%. Each recognition against a gallery with 1000 faces only needs about *3.05* seconds. These two experimental results together with the high performance recognition on partial faces demonstrate that our algorithm is not only effective but also efficient.

1 Introduction

With explicit shape information, three dimensional face recognition has been expected to overcome the problems, such as the variations of pose, lighting and expression [1,2], facing traditional 2D face recognition. Various techniques for 3D face recognition have been presented to make use of the shape clues [2]. However, as a relatively new research topic, a number of challenges still exist that limits the performance of current 3D face recognition algorithms both in accuracy and speed.

The first challenge is how to automatically extract the facial region from the raw data captured by a 3D scanner, which may contain hair, shoulder, and neck. Chang *et al.* [3] designed a skin model for the 2D color face image to help the 3D face extraction. Accurate registration between the image pixels and the 3D points are needed and the 3D facial region is found according to the skin detection. The requirement of an additional 2D image limits the usage of the approach.

The second challenge is the precise and fast alignment between two face models. A popular algorithm is the *iterative closest point* (ICP) [4] which is widely

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 603–616, 2008.

used during the matching stage in 3D face recognition [5,7,1,3]. The ICP algorithm iteratively minimizes the mean square distance (MSD) metric and has relatively good recognition performance. However, it suffers from facial surface distortion due to expression variation and noise [3]. Furthermore, the iterative process makes the ICP computationally expensive and the registration must be done for each model in the gallery. Consequently, it is not suitable for a recognition task with a large dataset.

The third problem is how to measure the similarity between two given facial shapes. Existing features include curvature [8], profile [9], surface descriptors [16], *Point Signature* [7], and *Spherical Face Representation* [17]. Some recent methods treat the aligned face model as a point set. The similarity is calculated via *Hausdorff distance* [5] or *Root-Mean-Square* (RMS) of the closest distances [3,10]. This kind of distances based on averaging provides a plain dissimilarity measure. The performance may decrease seriously in the presence of intrapersonal variation.

Expression variation is the fourth challenge. To reduce the effect of expressions, one approach is to choose only rigid regions for matching [7,3]. Another is to map or deform the original facial data to a middle model in which the distortion is reduced [10,11,6]. However, in the former case, there may not be such parts of the face that is shape invariant with sufficient discirminating power [2]. The latter work is interesting and improves the performance to some extent, but it is computationally demanding and more details are needed to understand the underlying mechanism.

The last challenge concerns computational efficiency. More information in 3D facial model leads to more computational cost. To our knowledge, there are no effective algorithms that can run in real time or close to real time in 3D face recognition when thousands of faces are in the gallery.

This paper proposes a new 3D face recognition approach named *Collective Shape Difference Classifier* (CSDC) to deal with the problems described above. The experimental results on FRGC v2 dataset [13], including 466 persons and 4007 models, achieve the rank-1 recognition rate *98.22%*, *which outperforms the best published approaches and its speed is nearly real-time*. The main contributions of our work are listed as follows:

(1) A fast and effective face posture alignment technique is presented to place all face models to a standard position and orientation. We show that this feature-based alignment step is enough for our later processing and no other iterative registration is needed. Also, the alignment is self-dependent which greatly reduces the time cost in face matching. This technique is to overcome the second and fifth challenges.

(2) A *Pure Shape Difference Map* (PSDM) is defined between two depth images sampled from two aligned 3D face models. The PSDM removes the alignment error in the pitch angle elegantly so as to encode more shape difference information between two faces. The scheme is used to tackle the second and the third challenges.

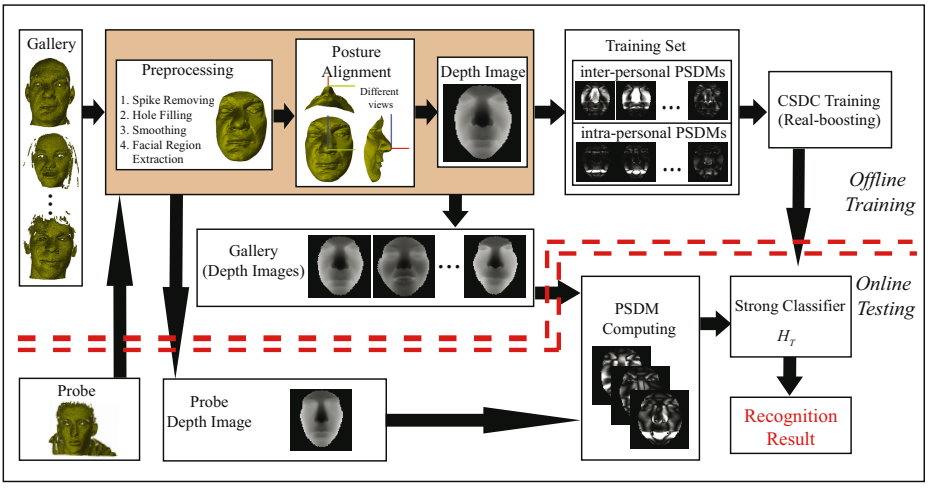


Fig. 1. The framework of our method

(3) The PSDM helps convert the multi-class face recognition problem to a 2-class classification problem, i.e., inter-personal and intra-personal classes, similar to the case in 2D face recognition [19,20]. Clearly, the different parts of PSDM do not contribute the same discriminability due to non-rigid distortion on the face. From the training inter-personal and intra-personal PSDM sets, the real-boosting [12] is used to choose the most discriminative local shape difference to build a strong classifier, namely, the CSDC. This part is to conquer the third, fourth, and fifth challenges.

Moreover, we also introduce a scheme for facial region extraction which addresses the first difficulty. Summarizing the above techniques, we show the whole framework in Fig. 1. Besides the accuracy and speed, another advantage of our method is its robustness on the partially missing face data. The specific techniques are discussed in the rest of this paper.

2 Posture Alignment

It is not clear how to define exact alignment of poses for two given 3D faces with rather different shapes. ICP uses the minimum of the *mean square distance* as a measure for alignment. We prefer to the coincidence of the prominent features such as the nose tips and the normals of the 3D face symmetry planes for alignment due to consideration of algorithmic efficiency and the fact that these two features (the nose tip and the normal) are relatively stable. By the two features, five out of the six degrees of freedom in face model can be fixed and we only need another point to determine the pitch angle. The top of the nose bridge is used for this task.

More detailed geometric definitions of the nose tip and the top of the nose bridge are necessary. Let the central profile curve C be the intersection of the

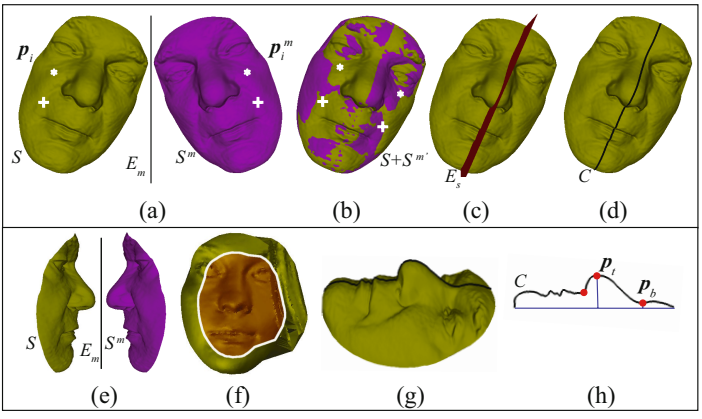


Fig. 2. Symmetry plane and profile finding. (a) An original model S and its mirror S^m. (b) Registration of S^m to S. (c) The symmetry plane E_s. (d) The profile C. (e) A special case of the mirror plane. (f) Points inside the closed white curve used for registration. (g) The horizontally placed profile. (h) The profile C, $\boldsymbol{p}_t$ and $\boldsymbol{p}_b$.

symmetry plane and the facial surface. Then, the nose tip $\boldsymbol{p}_t$ is defined as the point on the central profile C with the maximum distance to the line l_e passing through the two endpoints of C. The definition of the top of the nose bridge $\boldsymbol{p}_b$ is given in Section 2.2.

If the input facial surface only covers the facial region as shown in Fig. 2, these features can be found reliably by our method described in Section 2.1 and 2.2. However, failure can appear when the data contain the neck or much hair. Thus, the facial model should cover only the main facial region which does not exceed much of the forehead and the chin. Fortunately, it is not a difficult requirement. The facial region extraction introduced in Section 4.1 works well for this. The rest of this section discusses how to find the symmetry plane, the central profile, the nose tip and the top of the nose bridge, and how to align the facial model.

2.1 Facial Central Profile Finding

Let $S = \{\boldsymbol{p}_i \mid \boldsymbol{p}_i = (x_i, y_i, z_i), 1 \leq i \leq n\}$ denote the point set of a 3D facial model, and $S^m = \{\boldsymbol{p}_i^m \mid \boldsymbol{p}_i^m = (x_i^m, y_i^m, z_i^m), 1 \leq i \leq n\}$ be its mirror set with respect to some plane E_m, where the correspondence is naturally set up. Then we register S^m to S using the ICP algorithm [4] with S fixed (see Fig. 2(b)). After the registration, S_m becomes another set $S^{m\prime} = \{\boldsymbol{p}_i^{m\prime} \mid \boldsymbol{p}_i^{m\prime} = (x_i^{m\prime}, y_i^{m\prime}, z_i^{m\prime}), 1 \leq i \leq n\}$. The facial symmetry plane E_s is defined as the best fitting plane of the set of points $B = \{\boldsymbol{p}_i^b \mid \boldsymbol{p}_i^b = (\boldsymbol{p}_i + \boldsymbol{p}_i^{m\prime})/2, 1 \leq i \leq n\}$. The central profile C can easily be found by computing the intersection between S and E_s.

There are two issues that need to be addressed in the implementation of our technique. The first is that the mirror plane E_m should be chosen carefully. In some cases, an arbitrary E_m may cause the ICP to be nonconvergent, such as the

case shown in Fig. 2(e), because the initial poses of the two faces are changed too much. To deal with this problem, we propose the following scheme: 1) Perform PCA on S to obtain three new principal directions (eigenvectors) $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$ with their corresponding eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Roughly speaking, $\boldsymbol{v}_1$ is in the direction passing through the nose bridge, $\boldsymbol{v}_3$ is perpendicular to the front face, and $\boldsymbol{v}_2$ is perpendicular to both $\boldsymbol{v}_1$ and $\boldsymbol{v}_3$. 2) E_m is chosen as the plane passing through the centroid of S and with its normal being $\boldsymbol{v}_2$. Such a mirror plane E_m passes through S, making S^m and S already quite coincident, which leads to fast convergence of the ICP.

The second issue comes from the fact that our method is based on the mirror symmetry of human faces. However, the extracted facial region may not be so ideal, especially along the boundary. One example is given in Fig. 2(f). To guarantee the better convergence and alignment with ICP, a simple strategy is used where the points close to the boundary are discarded in the registration. When the facial model is represented as a 3D mesh, the inner points can easily be determined.

2.2 The Standard Coordinate Frame

According to definition, the nose tip $\boldsymbol{p}_t$ can be obtained from the central profile C by

$$\boldsymbol{p}_t = (x_t, y_t, z_t) = \arg\max_{\boldsymbol{p}_i^c \in C} dist_1(\boldsymbol{p}_i^c, l_e), \tag{1}$$

where $dist_1(\cdot, \star)$ is the Euclidean distance from a point to a line segment, and l_e is the line passing through the two endpoints of C.

Along the profile C with C placed horizontally as shown in Fig. 2(h), we find a local minimum point on each side of $\boldsymbol{p}_t$, which is closest to $\boldsymbol{p}_t$, denoted as $\boldsymbol{p}_1^*$ and $\boldsymbol{p}_2^*$. Then the top of the nose bridge $\boldsymbol{p}_b$ is defined as

$$\boldsymbol{p}_b = \arg\max_{\boldsymbol{p}^* \in \{\boldsymbol{p}_1^*, \boldsymbol{p}_2^*\}} dist_2(\boldsymbol{p}^*, \boldsymbol{p}_t), \tag{2}$$

where $dist_2(\cdot, \cdot)$ denotes the Euclidean distance between two points. Figs. 2(g) and (h) show the geometric relations of these new terms. It is worth noting that in the implementation for robustly finding the local minimum $\boldsymbol{p}_1^*$ or $\boldsymbol{p}_2^*$, we use its six nearest points along C, three on its one side and three on its other side, to detect if it is a local minimum.

Now let the normal of the symmetry plane E_s be $\boldsymbol{v}_x{}'$. We define a candidate frame $(\boldsymbol{v}_x{}', \boldsymbol{v}_y{}', \boldsymbol{v}_z{}')$ with $\boldsymbol{p}_t$ being the origin and $\boldsymbol{v}_y{}' = \boldsymbol{p}_b - \boldsymbol{p}_t$, $\boldsymbol{v}_z{}' = \boldsymbol{v}_x{}' \otimes \boldsymbol{v}_y{}'$, where $\otimes$ denotes the cross product of two vectors. The geometry of this frame is illustrated in Fig. 3(a). When we find the majority of the data points have positive z coordinates, we multiply all the x and z coordinates of the data points by -1 so that $\boldsymbol{v}_x{}'$ is in the direction towards the left-hand side of the face.

Finally, the standard coordinate frame is defined as $(\boldsymbol{v}_x, \boldsymbol{v}_y, \boldsymbol{v}_z)$, which is obtained by rotating the candidate frame counterclockwise around the x axis by an angle α ($\alpha = 30°$ in our experiments), as shown in Fig. 3(b). Compared with the candidate frame, in this standard frame, we can reduce the number of coincident points when the 3D face data are projected to the $x - y$ plane

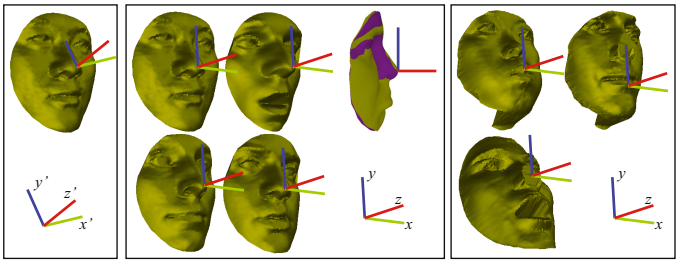


Fig. 3. (a) The candidate Frame. (b) Models in the standard frame with small pitch angle variation. (c) Some failure cases.

for constructing the PSDM (see the next section). Now we can align all models into this standard frame. Since different people have different nose shapes, the models, after alignment, may have small variation in the pitch angle (see Fig. 3 (b)). The effect of this pitch angle error can be removed with the PSDM, as described in the next section.

It should be mentioned that when there are many missing data and/or large distortion in a face model, it is possible for our method to align incorrectly. Three such examples are shown in Fig. 3(c). However, our method can obtain very good results for almost all the models in FRGC v2 and BU-3DFE databases.

3 Discriminative Local Shape Difference Boosting

Based on the aligned models, we investigate the shape differences and convert the 3D face recognition to a 2-class classification problem, i.e., the problem of determining a shape difference is inter-personal or intra-personal. In this section, we design a shape difference representation method, called *Pure Shape Difference Map* (PSDM), and a classifier, called *Collective Shape Difference Classifier* (CSDC). The PSDM aims to depict the shape difference/similarity between two models with reduced alignment error, and the CSDC can choose the most discriminative local patches from the PSDM to make the recognition decision.

3.1 Pure Shape Difference Map

Before obtaining the PSDM between two face models in the standard coordinate frame, we need to generate their depth images. By a sphere with radius r centered at the nose tip, the *region of interest* (ROI) is picked out and projected to a $w \times w$ image with the nose tip at the center of the image (we choose $w = r = 75$ in our experiments), as shown in Fig. 4(a). The positions that the projected face surface does not cover is set to a special value φ.

The difference image between two depth images can reflect the shape similarity between two 3D face models. However, the accuracy may be affected by the small alignment error in the pitch angle, which results from the small position change

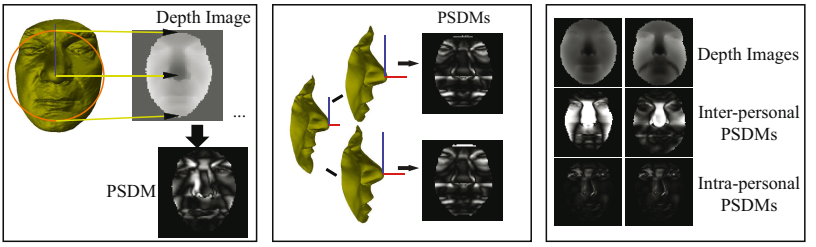


Fig. 4. (a) PSDM computing. (b) Two very similar PSDMs obtained with small alignment error in the pitch angles. (c) Examples of inter-personal and intra-personal PSDMs.

of the top of the nose bridge $\boldsymbol{p}_b$ among a group of 3D models. Suppose that each 3D model has a ground truth posture in the standard coordinate frame. Then we can see that the pitch angle variation of a face model brings approximately the same depth error on the same row of this depth image. Consequently, we can remove its effect by constructing a map of the shape difference, i.e., the PSDM.

Let I_1 (I_2) be a depth image and $I_1(i,j)$ ($I_2(i,j)$) be the depth value at the position (i,j), δ_i^1 (δ_i^2) be the depth error on row i, and the ground truth depth images corresponding to I_1 and I_2 be I_1^* and I_2^* respectively. Then,

$$I_1(i,j) = I_1^*(i,j) + \delta_i^1, \quad I_2(i,j) = I_2^*(i,j) + \delta_i^2. \tag{3}$$

The signed difference image D_s of I_1 and I_2 and the signed difference image D_s^* of I_1^* and I_2^* are defined as

$$D_s(i,j) = \begin{cases} I_1(i,j) - I_2(i,j), & \text{if } I_1(i,j) \neq \varphi \text{ and } I_2(i,j) \neq \varphi \\ \xi, & \text{otherwise} \end{cases}, \tag{4}$$

$$D_s^*(i,j) = \begin{cases} I_1^*(i,j) - I_2^*(i,j), & \text{if } I_1^*(i,j) \neq \varphi \text{ and } I_2^*(i,j) \neq \varphi \\ \xi, & \text{otherwise} \end{cases}, \tag{5}$$

where ξ is a special value denoting that $D_s(i,j)$ or $D_s^*(i,j)$ is invalid at the positions where at least one of the two depths under study equals φ. The PSDM, D_{ps} is defined as

$$D_{ps}(i,j) = \begin{cases} \mid D_s(i,j) - \frac{1}{k}\sum_{l=q}^{q+k-1} D_s(i,l) \mid, & \text{if } D_s(i,j) \neq \xi \text{ and} \\ & D_s(i,l) \neq \xi, q \leq l \leq q+k-1 \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

where $\{q, q+1, ..., q+k-1\}$ denotes k consecutive pixel positions on row i (In our experiments, we choose $q = 28$ and $k = 21$). Since $D_s(i,j) = D_s^*(i,j) + \delta_i^1 - \delta_i^2$, we have

$$D_{ps}(i,j) = \begin{cases} \mid D_s^*(i,j) - \frac{1}{k}\sum_{l=q}^{q+k-1} D_s^*(i,l) \mid, & \text{if } D_s(i,j) \neq \xi \text{ and} \\ & D_s(i,l) \neq \xi, q \leq l \leq q+k-1 \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

It can be seen from Eq. (7) that the alignment errors δ_i^1 and δ_i^2 are removed in $D_{ps}(i,j)$ (see Fig. 4(b)). Since the PSDM encodes the difference between two depth images with their alignment errors removed, we call it the *Pure Shape Difference Map* and use it as a critical representation for the recognition. Some examples of inter-personal and intra-personal PSDMs are shown in Fig. 4(c).

3.2 Collective Shape Difference Classifier

The PSDM keeps the information of the similarity between two face models and the Root-Mean-Square (RMS) is a choice for dissimilarity measure. However, the RMS runs into trouble when the noise and distortion of the facial surface occur. It is obvious that different parts of the PSDM have different contributions to recognition. Although we do not know which areas are the most discriminative across a broad range of distortion, the boosting algorithm can help select and combine them with suitable weights. This is the main idea of the *Collective Shape Difference Classifier* (CSDC).

The CSDC is a collective classifier of the form, $H_T(D_{ps}) = \sum_{t=1}^{T} c_t(D_{ps})$, where $c_t(D_{ps})$ is a weak classifier selected based on the simple features on the PSDMs during the real-boosting training [12], and T is the number of the weak classifiers. The output of $c_t(D_{ps})$ is a real value, i.e., confidence, and the final summed confidence is used as the similarity measure between the two 3D face models yielding D_{ps}.

In the learning of the CSDC, the intra-personal and inter-personal PSDMs are built from the given 3D face models, which compose the training set Q. Usually, the size of Q is very large mainly due to many different pairs of inter-personal depth images. It is impractical for a common PC to use all PSDMs in Q for training simultaneously. Thus, bootstrapping is used in learning by starting with all intra-personal and part of inter-personal PSDMs which form a subset Q_w of Q. Then we keep exchanging the inter-personal PSDMs between Q_w and Q so that all inter-personal samples can be used during the learning procedure. The detail of the learning is shown in the training part of Algorithm 1. Besides, two types of features are used in constructing the weak classifiers. One is the mean values of the rectangle patches in the PSDMs with different sizes. The other is Haar-like features that are frequently used in face detection [18]. Both use integrate images for better computational efficiency [18]. The testing part of Algorithm 1 shows how the CSDC carries out the recognition.

4 Experiments

Two 3D face databases, FRGC v2 [13] and BU-3DFE [14], are used to test our algorithm. The BU-3DFE database includes 100 persons and 2500 models. Each person has 7 kinds of expressions, 1 neutral and 6 other expressions. FRGC v2 has 466 persons and 4007 models.

Half of the 2400 models with non-neutral expressions in BU-3DFE are randomly selected and used together with the 100 neutral models to build the PSDM

Algorithm 1. Collective Shape Difference Classifier Training and Testing

Training Procedure:

Input:

1. $Q^{'}$ and $Q^{'}_{w}$: containing all and starting samples corresponding to Q and Q_w.
2. T: the target number of the weak classifiers.

Initialization: $w_{0,i}$

Learning:

For t = 1, 2, ..., T

1. Normalize the weights $w_{t,i}$.
2. Train weak classifiers on $Q^{'}_{w}$ and find the best weak classifier c_t.
3. Update the current collective classifier $H_t = H_{t-1} + c_t$.
4. If $sign(H_t)$ successfully classifies all samples in $Q^{'}_{w}$, update Q'_w by swapping 20% smallest weight inter-personal samples with those never used in Q'.
5. Update the weights $w_{t,i}$.

Output: H_T.

Testing Procedure:

Let $G = \{g_1, \ldots, g_r\}$ be the gallery set and M_p be a probe.

For i = 1, 2, ...,r

1. Compute D^i_{ps} between g_i and M_p,
2. Compute the score $\Omega_i = H_T(D^i_{ps})$.

Recognition result: Label$(M_p) = \arg\max_{1 \le i \le r}(\Omega_i)$

training set. The PSDMs are computed from pairs of these training models. Thus, 1200 intra-personal PSDMs and 118,800 inter-personal PSDMs are computed. These samples train the CSDC which is used for the recognition experiments. The remaining half of the models with non-neutral expressions in BU-3DFE are employed to determine the number of features in the CSDC, with the 100 neutral models forming the gallery set. For FRGC v2, the first session of the 466 persons are used as the gallery set, and the remaining 3541 models are used for testing. Note that we do not use any models in FRGC v2 for training. Besides, experiments on partial faces are also designed to test our algorithm.

Before testing, the models in FRGC v2 are smoothed and cropped by the methods given in Section 4.1. The models in BU-3DFE have been preprocessed by the providers. All models in these two databases are aligned to the standard coordinate system by our posture alignment method. The two parameters m and n in Algorithm 1 are 1200 and 10000, respectively.

4.1 Preprocessing

This subsection briefly discusses the preprocessing steps including face denoising and facial region extraction. The raw face data are assumed to be stored with known adjacency relations among the 3D points and the faces are placed

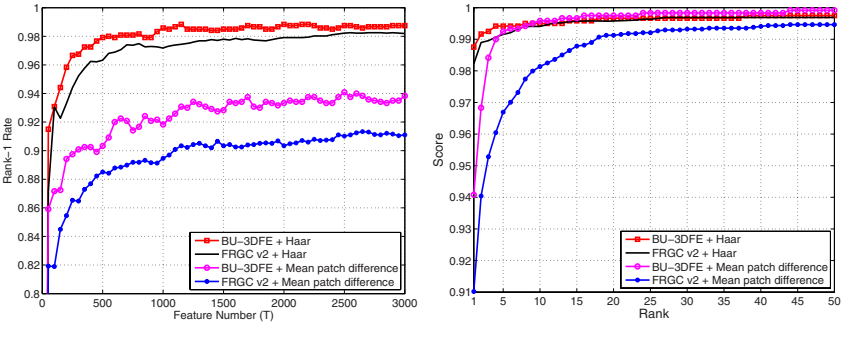
roughly in the common top-down posture (the front direction of the face can be arbitrary), as those in the FRGC v2 and 3D-BUFE databases. Many commercial 3D scanners can generate such data [2].

With this assumption, three Gaussian filters are designed to remove spikes, fill holes, and smooth the data with different variances. After that, the facial region is extracted based on the rough detection of the nose tip as follows: 1) 2/3 points close to the centroid of the denoised face in top-down direction are used to fit a plane E_1. Among the removed 1/3 points, 2/3 are at the bottom of the face and 1/3 at the top. The plane E_1 cuts the face data to two parts. The one with the smaller variance is selected as the candidate facial region. 2) With the points in the candidate facial region, we fit another plane E_2 and again select the part with the smaller variance. Among the points in this part, the point with the largest distance to the plane E_2 is selected as the approximate nose tip $\boldsymbol{p}_t{}'$. 3) By placing a sphere centered at $\boldsymbol{p}_t{}'$, the facial region can be cropped from the original denoised face (the sphere radius = 95 is selected in our experiments). This method is simple and fast, but works very well on the FRGC v2 database.

4.2 Effects of the Number and Type of Features

The number T of the weak classifiers in the CSDC balances the recognition accuracy and the running time both in training and testing. The maximum value $T = 3000$ is trained in our experiment. As shown in Fig. 5(a), the rank-1 rates keep increasing, but the curves become flat when $T > 2500$. Thus, we set $T = 2500$ in the subsequent experiments where the rank-1 rates on both databases exceed 98.2%.

As for the type of features, the mean values of rectangle patches give worse rank-1 rate than the Haar-like features by about 4% drop on the two sets. This result indicates that not only the shape difference but also its change patterns encode the similarity when intra-personal variations occur. With Haar-like features,



(a) (b)

Fig. 5. (a) Rank-1 recognition rates against the feature number. (b) Cumulative match characteristic curves (T = 2500).

recognition rates better than 99% are achieved after rank-3 in the *Cumulative Match Characteristic* (CMC) curves. (see Fig. 5(b)).

4.3 Comparison with Other Methods

We compare our CSDC with the ICP and the state of the art methods including the ARMS [3], the AFM [15], the GCD [10], and the *R3D* [17]. Each method uses all data in FRGC v2 database. For the AFM, the GCD, and our method, the first data session of each subject is used as the gallery set (total 466 faces) and the rest as probes (3541). The ARMS is with a superset, 449 vs. 3939 [3], and *R3D* chooses a neutral model for each person to compose a gallery and the remaining models are used as probes. The rank-1 rates obtained by the methods are given in Table 1, which clearly indicate that our CSDC performs better than the others. Note that the result of GCD is obtained from the authors of [10], and the others except ICP are quoted from the original papers.

Table 1. Comparison with five other works on FRGC v2

	ICP	ARMS	AFM	GCD	*R3D*	CSDC
Rank-1 Rate	75.66%	91.9%	97%	87.74%	96.2%	98.22%

4.4 Evaluation on Partial Faces

Two kinds of partial 3D faces are generated. One is obtained by shrinking the radius r used in projection for generating depth images (see Fig. 3.1), the other is by removing one or more quadrants of the face region. Some examples are shown in Fig. 6 and Fig. 7. Clearly, with such large parts of the face missing, our pose alignment may fail. However, here our purpose is to find out which parts of the face contribute more to the recognition, based on the partial faces that are aligned well.

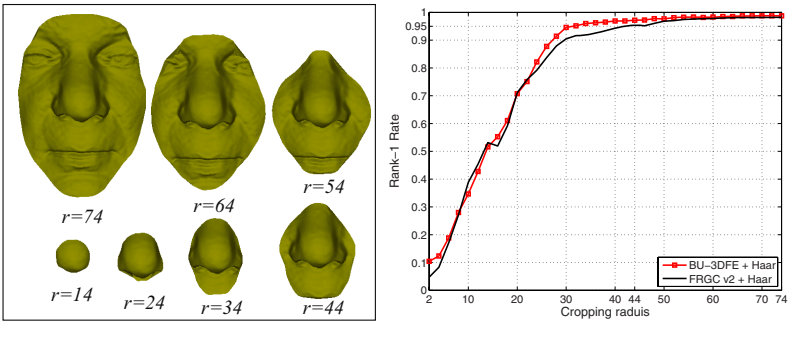


Fig. 6. Radius shrinking

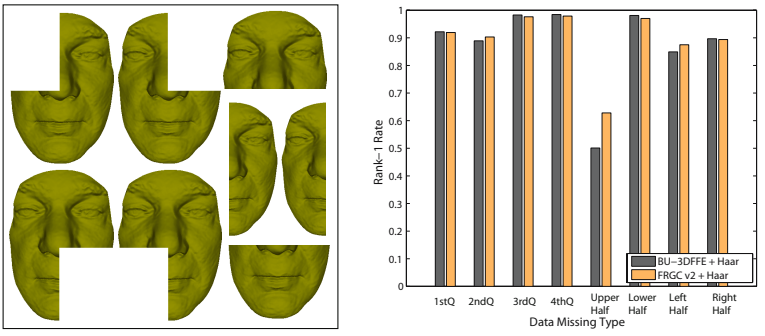


Fig. 7. Quadrant missing

Radius Shrinking. Totally 37 cropping radii, from 2 to 74 with step = 2, are tested in the experiment. Fig. 6 shows that the rank-1 rates are better than 95% for all $r \geq 44$ on both databases. It is rather surprising that such a small central patch as $r = 44$ still results in very good recognition rates by our CSDC. This is an important finding from our experiments and more attention should be paid to the nose region.

Quadrant Missing. We evaluate 8 kinds of quadrant missing. The results are illustrated in Fig. 7. The most important conclusion from this experiment is that the upper half of the face is more important than the lower half in recognition, with the average rank-1 rate 97% versus 56%.

4.5 Computational Performance

Although some of the previous methods can do verification nearly real-time, the computational performance of recognition is still a challenging task since the recognition must match the probe face against every gallery face. Thus the size of the gallery and the matching time are the main obstacle of fast recognition.

Usually, the running time of all steps, especially the preprocessing, depends on the number of points in 3D face models. We select the models with the minimum and maximum numbers of points from FRGC v2 to test our algorithm and also compute the average recognition time. The consumed time on a PC with CPU P4 3.0GHz and 2GB RAM is shown in Table 2.

The PSDM computation and the classification by the CSDC are very fast. In this experiment, since there are 466 models in the gallery, we need to compute

Table 2. Time used in each step

points number	Denoising	Facial Region Extraction	Posture Alignment	Depth Image	PSDMs (*466 times*)	Scores (*466 times*)
min. 53898	895ms	35ms	780ms	32ms	289ms	70ms
max. 197298	1844ms	130ms	1962ms	32ms	289ms	70ms
av. 100474	1195ms	71ms	978ms	32ms	289ms	70ms

466 PSDMs and 466 scores for classification (see Algorithm 1) for each probe. If the gallery has 1000 models, the average recognition time is about *3.05* seconds which is nearly real-time. Our method is several orders faster than existing methods.

5 Conclusion and Limitation

We have proposed an automatic 3D face recognition method which can obtain both high accuracy and computational efficiency. From the experimental results on the largest available public database, FRGC v2, the following conclusions can be drawn:

(1) The rank-1 rate better than 98% obtained by the CSDC indicates that the shape differences of 3D models are effective and the local area is critical. For most cases, our pose alignment is good enough.

(2) The low computational cost together with the accuracy makes our method practical for real time 3D face recognition system.

(3) Our method is robust on faces with large missing regions if the faces are aligned well. An important finding is that even a small nose area can give very good recognition results by the CSDC.

Although the CSDC works very well on common faces of approximate mirror-symmetry with a nose, it can fail when the data of the nose are missing, which causes incorrect alignment. This is the main limitation of our method. Fortunately, this is a rare case and most scanners can generate faces that can be handled by our algorithm.

Acknowledgement

This work was supported by a grant from the Research Grants Council of the Hong Kong SAR, China (Project No. CUHK 414306). Gang Pan was supported by Natural Science Foundation of China (No.60503019) and 863 Program of China (2008AA01Z149).

References

1. Medioni, G., Waupotitsch, R.: Face recognition and modeling in 3D. In: IEEE Int'l Workshop on AMFG (2003)
2. Chang, K., Bowyer, K., Flynn, P.: A Survey of Approaches and Challenges in 3D and Multi-Modal 2D+3D Face Recognition. Computer Vision and Image Understanding 101(1), 1–15 (2006)
3. Chang, K.I., Bowyer, K., Flynn, P.J.: Adaptive Rigid Multi-Region Selection for Handling Expression Variation in 3D Face Recognition. In: IEEE Workshop on FRGC (2005)
4. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. IEEE Trans. on PAMI 14(2), 239–256 (1992)

5. Russ, T.D., Koch, M.W., Little, C.Q.: A 2D range Hausdorff approach for 3D face recognition. In: IEEE Workshop on FRGC (2005)
6. Lu, X., Jain, A.K.: Deformation modeling for robust 3D face matching. In: IEEE Conf. on CVPR (2006)
7. Chua, C.S., Han, F., Ho, Y.K.: 3D Human Face Recognition Using Point Signature. In: Int'l Conf. on FG (2000)
8. Gordon, G.G.: Face recognition from depth maps and surface curvature. In: SPIE Conf. on Geometric Methods in Computer Vision (1991)
9. Wu, Y.J., Pan, G., Wu, Z.H.: Face Authentication based on Multiple Profiles Extracted from Range Data. In: Int'l Conf. on AVBPA (2003)
10. Wang, Y.M., Pan, G., Wu, Z.H.: 3D Face Recognition in the Presence of Expression: A Guidance-based Constraint Deformation Approach. In: IEEE Conf. on CVPR (2007)
11. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Three dimensional face recognition. Int'l Journal of Computer Vision 64(1), 5–30 (2005)
12. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. In: Annual Conf. on Computational Learning Theory (1998)
13. Phillips, P.J., Flynn, P.J., Scruggs, T., Bowyer, K.W., Chang, J., Hoffman, K., Marques, J., Min, J., Worek, W.: Overview of the Face Recognition Grand Challenge. In: IEEE Conf. on CVPR (2005)
14. Yin, L., Wei, X., Sun, Y., Wang, J., Rosato, M.: A 3D facial expression database for facial behavior research. In: Int'l Conf. on FG (2006)
15. Kakadiaris, I.A., Passalis, G., Toderici, G., et al.: Three-Dimensional Face recognition in the presence of facial expressions: An annotated deformable model approach. IEEE Trans. on PAMI 29(4), 640–649 (2007)
16. Moreno, A.B., Sanchez, A., Velez, J.F., et al.: Face recognition using 3D surface-extracted descriptors. In: Irish Machine Vision and Image Processing Conference (2003)
17. Mian, A., Bennamoun, M., Owens, R.: An Efficient Multimodal 2D-3D Hybrid Approach to Automatic Face recognition. IEEE Trans. on PAMI 29(11), 1927–1943 (2007)
18. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: IEEE Conf. on CVPR (2001)
19. Wang, X., Tang, X.: A unified framework for subspace face recognition. IEEE Trans. on PAMI 26(9), 1222–1228 (2004)
20. Wang, X., Tang, X.: Unified subspace analysis for face recognition. In: IEEE International Conference on Computer Vision (2003)

Efficiently Learning Random Fields for Stereo Vision with Sparse Message Passing

Jerod J. Weinman[1], Lam Tran[2], and Christopher J. Pal[2]

[1] Dept. of Computer Science
Grinnell College
Grinnell, IA 50112
weinman@grinnell.edu
[2] Dept. of Computer Science
University of Rochester
Rochester, NY 14627
{ltran,cpal}@cs.rochester.edu

Abstract. As richer models for stereo vision are constructed, there is a growing interest in learning model parameters. To estimate parameters in Markov Random Field (MRF) based stereo formulations, one usually needs to perform approximate probabilistic inference. Message passing algorithms based on variational methods and belief propagation are widely used for approximate inference in MRFs. Conditional Random Fields (CRFs) are discriminative versions of traditional MRFs and have recently been applied to the problem of stereo vision. However, CRF parameter training typically requires expensive inference steps for each iteration of optimization. Inference is particularly slow when there are many discrete disparity levels, due to high state space cardinality. We present a novel CRF for stereo matching with an explicit occlusion model and propose sparse message passing to dramatically accelerate the approximate inference needed for parameter optimization. We show that sparse variational message passing iteratively minimizes the KL divergence between the approximation and model distributions by optimizing a lower bound on the partition function. Our experimental results show reductions in inference time of one order of magnitude with no loss in approximation quality. Learning using sparse variational message passing improves results over prior work using graph cuts.

1 Introduction

There has been growing interest in creating richer models for stereo vision in which more parameters are introduced to create more accurate models. In particular, recent activity has focused on explicitly accounting for occlusions in stereo vision models. For example, Kolmogorov and Zabih [1] have directly incorporated occlusion models in an energy function and graph cuts minimization framework. Sun et al. [2] explored a symmetric stereo matching approach whereby they: (1) infer the disparity map in one view considering the occlusion map of the other view and (2) infer the occlusion map in one view given the

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 617–630, 2008.

disparity map of the other view. More recently, Yang et al. [3] have achieved impressive results building on the dual image set-up and using color weighted correlations for patch matching. They found that this approach made match scores less sensitive to occlusion boundaries as occlusions often cause color discontinuities. As all of these methods involve creating richer models to obtain greater disparity accuracy, there is a growing need to learn or estimate model parameters in an efficient and principled way.

In contrast to previous work [1][3], we are interested in developing a completely probabilistic formulation for stereo with occlusions modeled as additional states of random variables in a conditional random field (CRF). As noted by Yang et al. [3], more studies are needed to understand the behavior of algorithms for optimizing parameters in stereo models. For example, they note that their approach might be re-formulated in an expectation maximization framework. One goal of this paper is to address these types of questions in a general way. As we will show, when traditional stereo techniques are augmented with an occlusion model and cast in a CRF framework, learning can be achieved via maximum (conditional) likelihood estimation. However, learning becomes more challenging as the stereo images and probabilistic models become more realistic.

Belief propagation (BP) [4] and variational methods [5] are widely used techniques for inference in probabilistic graphical models. Both techniques have been used for inference and learning in models with applications ranging from text processing to computer vision [6,7]. Winn and Bishop proposed Variational Message Passing (VMP) [8] as a way to view many variational inference techniques, and it represents a general purpose algorithm for approximate inference. The approach is similar in nature to BP in that messages propagate local information throughout a graph, and the message computation is similar. However, VMP optimizes a lower bound on the log probability of observed variables in a generative model.

Experimental and theoretical analysis of variational methods has shown that while the asymptotic performance of other methods such as sampling [9] can be superior, frequently variational methods are faster for approximate inference. However, many real world problems require models with variables having very large state spaces. Under these conditions, inference with variational methods becomes very slow, diminishing any gains. We address this by proposing *sparse variational methods*. These methods also provide theoretical guarantees that the Kullback-Leibler (KL) divergence between approximate distributions and true distributions are iteratively minimized. Previous work by Pal et al. [10] explored sparse methods for approximate inference using BP in chain-structured graphs. Unlike varational inference, in loopy models BP does not have a direct connection to the probability of data under a model. The method we propose here combines the theoretical benefits of variational methods with the time-saving advantages of sparse messages.

In this work, we use a lattice-structured CRF for stereo vision. This leads to energy functions with a traditional form—single variable terms and pairwise terms. Importantly, unlike purely energy-based formulations [1], since we cast

the stereo problem as a probability distribution, we are able to view approximate inference and learning in the model from the perspective of variational analysis. While we focus upon approximate inference and learning in lattice-structured conditional random fields [11] applied to stereo vision, our theoretical results and some experimental insights are applicable to CRFs, MRFs and Bayesian Networks with arbitrary structures.

Many techniques have been used for parameter learning in CRFs used for image labeling, such as pseudo-likelihood [12], tree-based reparameterization (TRP) [13], and contrastive divergence [14]. Pseudo-likelihood is known to give poor estimates of interaction parameters, especially in conditional models. TRP is a variant of BP and has the same potential drawbacks. Contrastive divergence uses MCMC but does not require convergence to equilibrium for approximating the model likelihood gradients used for learning. However, models for image labeling usually only have a few states, whereas the state space in stereo models is much larger, for the many possible disparities. Thus, we believe that the sparse learning techniques we propose here will be an important contribution, providing the additional theoretical guarantees of variational methods.

Previous efforts at learning parameters for stereo models have used graph cuts to provide point estimates [15]. While recent work has shown that sequential tree-reweighted max-product message passing (TRW-S) has the ability to produce even better minimum energy solutions than graph cuts [16], max-product TRW-S also produces point estimates as opposed to approximate marginals.

The remainder of the paper is structured as follows. In section 2, we present a canonical conditional random field for the stereo vision problem. The canonical model is then augmented to explicitly account for occlusions. Next, we show how approximate inference is used for learning and to infer depth in an image. Section 3 shows how sparse variational message passing minimizes the KL divergence between a variational approximation and a distribution of interest. Results comparing sparse BP and VMP with graph cuts are given in section in section 4. Using variational distributions for learning improves results over the point estimate given by graph cuts, and sparse message passing can lead to an order of magnitude reduction in inference time. Furthermore, we show how learning parameters with our technique allows us to improve the quality of occlusion predictions in more richly structured CRFs.

2 Stereo Vision and CRFs

The stereo vision problem is to estimate the *disparity* (horizontal displacement) at each pixel given a rectified pair of images. It is common in MRF-based stereo vision methods to work with energy functions of the form

$$F(\boldsymbol{x}, \boldsymbol{y}) = \sum_i U(x_i, \boldsymbol{y}) + \sum_{i \sim j} V(x_i, x_j, \boldsymbol{y}) \tag{1}$$

where U is a *data term* that measures the compatibility between a disparity x_i and observed intensities $\boldsymbol{y}$, and V is a *smoothness term* between disparities at neighboring locations $i \sim j$ [17].

We construct a formal CRF probability model for stereo by normalizing the exponentiated F over all values for $\boldsymbol{X}$,

$$P(\boldsymbol{X} \mid \boldsymbol{y}) = \frac{1}{Z(\boldsymbol{y})} \exp(-F(\boldsymbol{X}, \boldsymbol{y})) \quad \text{with} \quad Z(\boldsymbol{y}) = \sum_{\boldsymbol{x}} \exp(-F(\boldsymbol{x}, \boldsymbol{y})). \quad (2)$$

The normalizer $Z(\boldsymbol{y})$ is typically referred to as the partition function.

2.1 A Canonical Stereo Model

The CRF of (2) is a general form. Here we present the specific CRF used for our experiments on stereo disparity estimation in section 4, following the model proposed by Scharstein and Pal [15]. The data term U simply measures the absolute intensity difference between corresponding pixels, summed over all color bands. We use the measure of Birchfield and Tomasi [18] for invariance to image sampling. The smoothness term V is a gradient-modulated Potts model [17,15] with $K = 3$ parameters:

$$V(x_i, x_j, \boldsymbol{y}) = \begin{cases} 0 & \text{if } x_i = x_j \\ \theta_k & \text{if } x_i \neq x_j \text{ and } g_{ij} \in B_k \end{cases} \quad (3)$$

Here g_{ij} is the color gradient between neighboring pixels i and j, and the B_k are three consecutive gradient bins with interval breakpoints from the set $\{0, 4, 8, \infty\}$. Let Θ_v denote all the parameters.

2.2 Occlusion Modeling

To account for occlusion, we create a model with an explicit occlusion state for the random variable associated with each pixel in the image. In our extended model $x_i \in \{0, \ldots, N-1\} \vee$ "occluded". The local data term U in our extended model has the form:

$$U(x_i, \boldsymbol{y}) = \begin{cases} c_i(x_i) & \text{if } x_i \neq \text{"occluded"} \\ \theta_o & \text{if } x_i = \text{"occluded"}, \end{cases} \quad (4)$$

where $c_i(x_i)$ is the Birchfield and Tomasi cost for disparity x_i at pixel i, as before. The new parameter θ_o is a local bias for predicting the pixel to be occluded.

We may also extend the gradient modulated smoothness terms to treat occluded states with a separate set of parameters such that:

$$V(x_i, x_j, \boldsymbol{y}) = \begin{cases} 0 & \text{if } x_i = x_j \text{ and } x_i \neq \text{"occluded"} \\ \theta_k & \text{if } x_i \neq x_j,\ g_{ij} \in B_k \text{ and both } x_i, x_j \neq \text{"occluded"} \\ \theta_{o,o} & \text{if } x_i = x_j \text{ and } x_i = \text{"occluded"} \\ \theta_{o,k} & \text{if } x_i \neq x_j,\ g_{ij} \in B_k \text{ and } x_i \text{ or } x_j = \text{"occluded"}. \end{cases} \quad (5)$$

2.3 Parameter Learning

Since the function $F(\boldsymbol{x}, \boldsymbol{y})$ is parameterized by $\Theta = (\theta_o, \Theta_v)$, these parameters may be learned in a maximum-likelihood framework with labeled training pairs. The objective function and gradient for one training pair $(\boldsymbol{x}, \boldsymbol{y})$ is

$$\mathcal{O}(\Theta) = \log P(\boldsymbol{x} \mid \boldsymbol{y}; \Theta) \tag{6}$$

$$= -F(\boldsymbol{x}, \boldsymbol{y}; \Theta) - \log Z(\boldsymbol{y}) \tag{7}$$

$$\nabla\mathcal{O}(\Theta) = -\nabla F(\boldsymbol{x}, \boldsymbol{y}; \Theta) + \langle \nabla F(\boldsymbol{x}, \boldsymbol{y}; \Theta) \rangle_{P(\boldsymbol{X} \mid \boldsymbol{y}; \Theta)} . \tag{8}$$

The particular factorization of $F(\boldsymbol{x}, \boldsymbol{y})$ in (1) allows the expectation in (8) to be decomposed into a sum of expectations over gradients of each term $U(x_i, \boldsymbol{y})$ and $V(x_i, x_j, \boldsymbol{y})$ using the corresponding marginals $P(X_i \mid \boldsymbol{y}; \Theta)$ and $P(X_i, X_j \mid \boldsymbol{y}; \Theta)$, respectively.

In previous work [15], graph cuts was used to find the most likely configuration of $\boldsymbol{X}$. This was taken as a point estimate of $P(\boldsymbol{X} \mid \boldsymbol{y}; \Theta_v)$ and used to approximate the gradient. Such an approach is potentially problematic for learning when the marginals are multi-modal or diffuse and unlike a delta function. Fortunately, a variational distribution $Q(\boldsymbol{X})$ can provide more flexible approximate marginals that may be used to approximate the gradient. We show in our experiments that using these marginals for learning is better than using a point estimate in situations when there is greater uncertainty in the model.

3 CRFs and Sparse Mean Field

In this section we derive the equations for *sparse* mean field inference using a variational message passing (VMP) perspective [8]. We show that sparse VMP will iteratively minimize the KL divergence between an approximation Q and the distribution P. Furthermore, we present sparse VMP in the context of CRFs and show that the functional we optimize is an upper bound on the negative log conditional partition function.

3.1 Mean Field

Here we briefly review the standard mean field approximation for a conditional distribution like (2). Let X_i be a discrete random variable taking on values x_i from a finite alphabet $\mathcal{X} = \{0, \ldots, N-1\}$. The concatenation of all random variables $\boldsymbol{X}$ takes on values denoted by $\boldsymbol{x}$, and the conditioning observation is $\boldsymbol{y}$. Variational techniques, such as mean field, minimize the KL divergence between an approximate distribution $Q(\boldsymbol{X})$ and the true distribution $P(\boldsymbol{X} \mid \boldsymbol{y})$. For the conditional distribution (2), the divergence is

$$\begin{aligned} \mathrm{KL}(Q(\boldsymbol{X}) \parallel P(\boldsymbol{X} \mid \boldsymbol{y})) &= \sum_{\boldsymbol{x}} Q(\boldsymbol{x}) \log \frac{Q(\boldsymbol{x})}{P(\boldsymbol{x} \mid \boldsymbol{y})} \\ &= \sum_{\boldsymbol{x}} Q(\boldsymbol{x}) \log \frac{Q(\boldsymbol{x}) Z(\boldsymbol{y})}{\exp(-F(\boldsymbol{x}, \boldsymbol{y}))} \\ &= \langle F(\boldsymbol{x}, \boldsymbol{y}) \rangle_{Q(\boldsymbol{X})} - H(Q(\boldsymbol{X})) + \log Z(\boldsymbol{y}) . \end{aligned} \tag{9}$$

The energy of a configuration $\boldsymbol{x}$ is $F(\boldsymbol{x}, \boldsymbol{y})$. We define a "free energy" of the variational distribution to be

$$\mathcal{L}(Q(\boldsymbol{X})) = \langle F(\boldsymbol{x}, \boldsymbol{y}) \rangle_{Q(\boldsymbol{X})} - H(Q(\boldsymbol{X})). \tag{10}$$

Thus, the free energy is the expected energy under the variational distribution $Q(\boldsymbol{X})$, minus the entropy of Q. The divergence then becomes

$$\mathrm{KL}(Q(\boldsymbol{X}) \parallel P(\boldsymbol{X} \mid \boldsymbol{y})) = \mathcal{L}(Q(\boldsymbol{X})) + \log Z(\boldsymbol{y}). \tag{11}$$

Since the KL divergence is always greater than or equal to zero, it holds that

$$\mathcal{L}(Q(\boldsymbol{X})) \geq -\log Z(\boldsymbol{y}), \tag{12}$$

and the KL divergence is minimized at zero when the free energy equals the negative log partition function. Since $\log Z(\boldsymbol{y})$ is constant for a given observation, minimizing the free energy serves to minimize the KL divergence.

Mean field updates will minimize $\mathrm{KL}(Q(\boldsymbol{X}) \parallel P(\boldsymbol{X} \mid \boldsymbol{y}))$ for a factored distribution $Q(\boldsymbol{X}) = \prod_i Q(X_i)$. Using this factored Q, we can express our objective as

$$\begin{aligned} \mathcal{L}(Q(\boldsymbol{X})) &= \sum_{\boldsymbol{x}} \prod_i Q(x_i) F(\boldsymbol{x}, \boldsymbol{y}) + \sum_i \sum_{x_i} Q(x_i) \log Q(x_i) \\ &= \sum_{\boldsymbol{x}} Q(x_j) \langle F(\boldsymbol{x}, \boldsymbol{y}) \rangle_{\prod_{i:i \neq j} Q(X_i)} - H(Q(X_j)) - \sum_{i:i \neq j} H(Q(X_i)), \end{aligned} \tag{13}$$

where we have factored out the approximating distribution $Q(X_j)$ for one variable, X_j. We form a new functional by adding Lagrange multipliers to constrain the distribution to sum to unity. This yields an equation for iteratively calculating an updated approximating distribution $Q^*(x_j)$ using the energy F and the distributions $Q(X_i)$ for other i:

$$Q^*(x_j) = \frac{1}{Z_j} \exp\left(- \langle F(\boldsymbol{x}, \boldsymbol{y}) \rangle_{\prod_{i:i \neq j} Q(X_i)} \right), \tag{14}$$

where Z_j is a normalization constant computed for each update so that $Q^*(x_j)$ sums to one. See Weinman et al. [19] for the complete derivation of (14). Iteratively updating $Q(X_j)$ in this manner for each variable X_j will monotonically decrease the free energy $\mathcal{L}(Q(\boldsymbol{X}))$, thus minimizing the KL divergence.

3.2 Sparse Updates

Variational marginals can be more valuable than graph cuts-based point estimates for accurate learning or other predictions. However, when the state space of the X_j is large, calculating the expectations within the mean field update (14) can be computationally burdensome. Here we show how to dramatically reduce the computational load of calculating updates when many states have a very low (approximate) probability. The sparse methods presented here represent a middle way between a fully-Bayesian approach and a simple point estimate. While the former

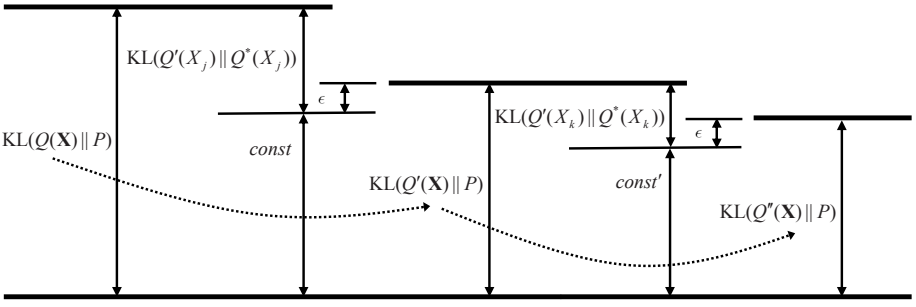


Fig. 1. Minimizing the global KL divergence via two different sparse local updates. The *global* divergence $\mathrm{KL}(Q(\mathbf{X}) \parallel P)$ can be decomposed into a *local* update plus a constant: $\mathrm{KL}(Q^{'}(X_j) \parallel Q^{*}(X_j)) + const.$ Consequently, at each step of sparse variational message passing we may minimize different local divergences to within some ϵ and when updating different local Qs, we minimize the global KL divergence.

considers all possibilities with their corresponding (often small) probabilities, the latter only considers the most likely possibility. Sparse updates provide a principled method for retaining an arbitrary level of uncertainty in the approximation.

The idea behind the sparse variational update is to eliminate certain values of x_j from consideration by making their corresponding variational probabilities $Q(x_j)$ equal to zero. Such zeros make calculating the expected energy for subsequent updates substantially easier, since only a few states must be included in the expectation. The eliminated states are those with low probabilities to begin with. Next we show how to bound the KL divergence between the original and sparse versions of $Q(X_j)$.

Given (11), (14), and (14) we can express $\mathrm{KL}(Q(\boldsymbol{X}) \parallel P(\boldsymbol{X} \mid \boldsymbol{y}))$ as a function of a *sparse* update $Q^{'}(X_j)$, the original mean field update $Q^{*}(X_j)$ and the other $Q(X_i)$, where $i \neq j$:

$$\begin{aligned}\mathrm{KL}(Q(\boldsymbol{X}) \parallel P(\boldsymbol{X} \mid \boldsymbol{y})) = \mathrm{KL}(Q'(X_j) \parallel Q^{*}(X_j)) \\ + \log Z_j + \log Z(\boldsymbol{y}) - \sum_{i:i\neq j} H(Q(X_i)).\end{aligned} \tag{15}$$

Since the last three terms of (15) are constant with respect to our update $Q^{'}(X_j)$, $\mathrm{KL}(Q(\boldsymbol{X}) \parallel P(\boldsymbol{X} \mid \boldsymbol{y}))$ is minimized when $Q^{'}(X_j) = Q^{*}(X_j)$. At each step of *sparse* variational message passing, we will minimize $\mathrm{KL}(Q^{'}(\boldsymbol{X}_j) \parallel Q^{*}(\boldsymbol{X}_j))$ to within some small ϵ. As a result, each update to a different $Q(\boldsymbol{X}_j)$ yields further minimization of the *global* KL divergence. These relationships are illustrated in Figure 1.

If each X_j is restricted to a subset of values $x_j \in \mathcal{X}_j \subseteq \mathcal{X}$, we may define sparse updates $Q^{'}(X_j)$ in terms of the original update $Q^{*}(X_j)$ and the characteristic/indicator function $\mathbf{1}_{\mathcal{X}_j}(x_j)$ for the restricted range:

$$Q'(x_j) = \frac{\mathbf{1}_{\mathcal{X}_j}(x_j)}{Z'_j} Q^{*}(x_j), \tag{16}$$

where the new normalization constant is

$$Z'_j = \sum_{x_j} Q'(x_j) = \sum_{x_j \in \mathcal{X}_j} Q^*(x_j). \tag{17}$$

Thus, the divergence between a sparse update and the original is

$$\begin{aligned} &\mathrm{KL}\left(Q'(X_j) \parallel Q^*(X_j)\right) \qquad (18)\\ &= \sum_x \frac{\mathbf{1}_{\mathcal{X}_j}(x_j)}{Z'_j} Q^*(x_j) \log\left(\left(\frac{\mathbf{1}_{\mathcal{X}_j}(x_j)}{Z'_j} Q^*(x_j)\right) \Big/ Q^*(x_j)\right) \\ &= -\log Z'_j \frac{1}{Z'_j} \sum_{x \in \mathcal{X}_j} Q^*(x_j) \\ &= -\log Z'_j. \qquad (19) \end{aligned}$$

As a consequence, it is straightforward and efficient to compute a maximally sparse $Q'(X_j)$ such that $\mathrm{KL}\left(Q'(X_j) \parallel Q^*(X_j)\right) \leq \epsilon$ by sorting the $Q^*(x_j)$ values and performing a sub-linear search to satisfy the inequality. For example, if we wish to preserve 99% of the probability mass in the sparse approximation we may set $\epsilon = -\log 0.99 \approx .01$. Figure 1 illustrates the way in which sparse VMP iteratively minimizes the $\mathrm{KL}\left(Q(\boldsymbol{X}) \parallel P(\boldsymbol{X} \mid \boldsymbol{y})\right)$ after each iteration of message passing. In section 4 we show how using sparse messages can yield a dramatic increase in inference speed.

4 Experiments

In this section we present the results of two sets of experiments. The first compares sparse and traditional mean field methods for approximate inference, showing how sparse message passing can greatly accelerate free energy minimization. The second compares the performance of models learned using approximate marginals from both sparse mean field and a point estimate of the posterior marginals from graph cuts.

As training and test data we use 6 stereo pair images with ground-truth disparities from the 2005 scenes of the Middlebury stereo database[1]. These images are roughly 450×370 pixels and have discretized disparities with $N = 80$ states. Thus, when there are more than 600,000 messages of length N to send in any round of mean field updates for one image, shortening these to only a few states for most messages can dramatically reduce computation time.

4.1 Inference

The variational distribution $Q(\boldsymbol{X})$ provides approximate marginals $Q(X_i)$ that may be used for computing an approximate likelihood and gradient for training. These marginals are also used to calculate the mean field updates during free

[1] `http://vision.middlebury.edu/stereo/data`

energy minimization. If these marginals have many states with very low probability, discarding them will have minimal effect on the update. First, we examine the need for sparse updates by evaluating the amount of uncertainty in these marginals. Then, we show how much time is saved by using sparse updates.

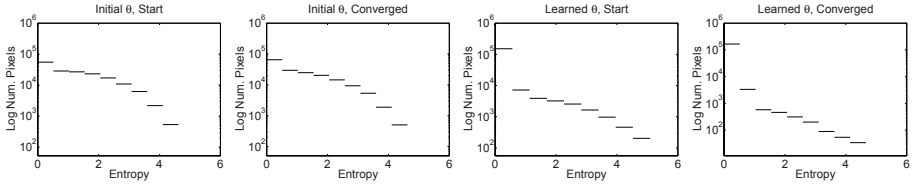


Fig. 2. Histograms of approximate marginal entropies $H\left(Q\left(X_i\right)\right)$ from the variational distributions for each pixel at the start (after the first round) of mean field updates and at their convergence; values using the initial and learned parameters Θ_v of the canonical model are shown

Our first set of experiments uses the simpler canonical stereo model having the smoothness term V of (3). Figure 2 shows histograms of the marginal entropies $H\left(Q\left(X_i\right)\right)$ during free energy minimization with two sets of parameters, the initial parameters, $\Theta_v = \mathbf{1}$, and the learned Θ_v. We initialize the variational distributions $Q\left(X_i\right)$ to uniform and perform one round of VMP updates. Although most pixels have very low entropy, the initial model still has several variables with 2-4 "nats" (about 3-6 bits) of uncertainty. Once the model parameters are learned, the marginal entropies after one round of mean field updates are much lower. By the time the mean field updates converge and free energy is minimized, only a small percentage (less than three percent) have more than a half nat (less than two bits) of uncertainty. However, if point estimates are used, the uncertainty in these marginals will not be well represented. Sparse messages will allow those variables with low entropy to use few states, even a point estimate, while the handful of pixels with larger entropy may use more states.

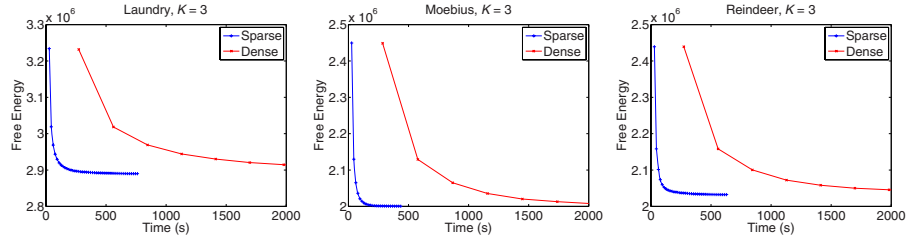


Fig. 3. Comparison of CPU time for free energy minimization with sparse and dense mean field updates using parameters Θ_v learned in the canonical model with three images (Art, Books, Dolls)

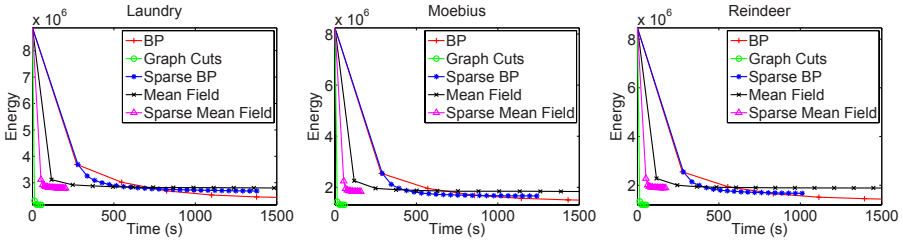


Fig. 4. CPU time versus energy for graph cuts, sum-product belief propagation, and mean field using parameters Θ_v learned with three images (Art, Books, Dolls). *Maximum posterior marginal* (MPM) prediction is used with the approximate marginal at each iteration.

The variational distribution has many states carrying low probability, even at the outset of training. We may greatly accelerate the update calculations by dropping these states according using (19) and our criterion. Figure 3 shows the free energy after each round of updates for both sparse and dense mean field. In all cases, sparse mean field has nearly reached the free energy minimum before one round of dense mean field updates is done. Importantly, the minimum free energy found with sparse updates is roughly the same as its dense counterpart.

As a comparison, we show in Figure 4 the true energy $F(\boldsymbol{x}, \boldsymbol{y})$ on several images during each iteration of several methods. It is important to note that only graph cuts explicitly minimizes this energy, but it is demonstrative of the relative speed and behavior of the methods.

4.2 Learning

As Figure 4 shows, graph cuts does a very good job of finding a minimum energy configuration. This is useful for making a prediction in a good model. However, maximizing the log likelihood (7) for learning requires marginals on the lattice. When the model is initialized, these marginals have higher entropy (Figure 2) representing the uncertainty in the model. At this stage of learning, the point estimate resulting from an energy minimization may not be a good approximation to the posterior marginals. In fact, using the graph cuts solution as a point estimate distribution having zero entropy, sparse mean field finds a lower *free* energy at the initial parameters $\Theta_v = \mathbf{1}$.

We compare the results of learning using two methods for calculating the gradient: sparse mean field and graph cuts. As demonstrated earlier, the model has the highest uncertainty at the beginning of learning. It is at this point when sparse mean field has the greatest potential for improvement over graph cuts.

For learning, we use a small initial step size and a simple gradient descent algorithm with an adaptive rate. For prediction evaluation, we use graph cuts to find the most probable labeling, regardless of training method. We use leave-one-out cross validation on the six images.

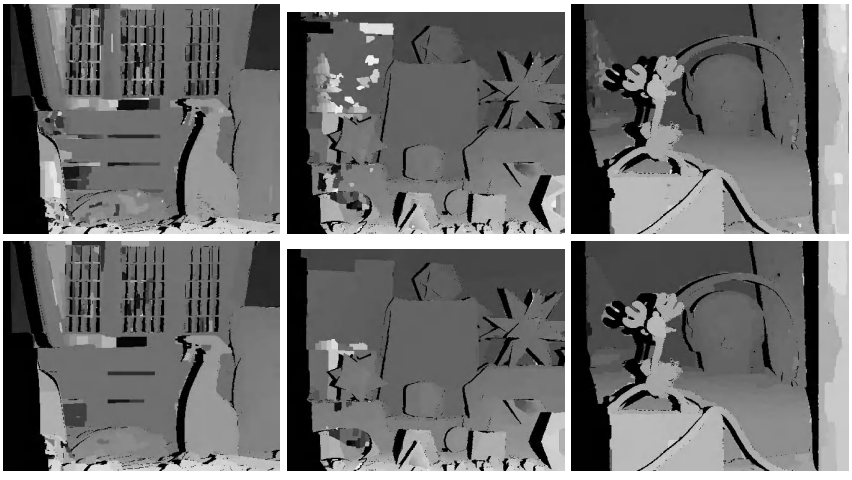


Fig. 5. Test images comparing prediction (using graph cuts) after one round of learning the canonical model with graph cuts (top) or sparse mean field (bottom). Occluded areas are black. Images (l-r): Laundry, Moebius, Reindeer.

After just one iteration, the training and test error with sparse mean field is markedly lower than that of the model trained with graph cuts for inference. Figure 5 shows the corresponding depth images after one iteration.

In Table 1, we compare the results of training using point estimates provided by graph cuts, as in previous work [15], and sparse mean field, the method proposed in this paper. We do not present results based on BP or dense mean field

Table 1. Comparison of learning with graph cuts and sparse mean field. The disparity error (percentage of incorrectly predicted pixels) given for the canonical stereo model and the gradient-modulated occlusion model (with Eqs. (4) and (5)). For the gradient-modulated occlusion model we show the occlusion prediction error (percentage).

Metric	Method	Art	Books	Dolls	Laundry	Moebius	Reindeer	Average
Canonical Model - leave-one-out training & testing								
Disparity Error	Graph Cuts	20.83	23.64	10.69	30.04	15.80	14.13	19.17
	Sparse Mean Field	17.70	23.08	10.67	29.16	15.43	13.37	18.22
Gradient-Modulated Occlusion Model - leave-one-out training & testing								
Disparity Error	Graph Cuts	21.82	24.10	11.94	27.54	11.08	16.74	19.30
	Sparse Mean Field	21.05	23.14	11.62	27.37	11.45	16.44	18.93
Occlusion Error	Graph Cuts	34.50	28.27	32.99	36.89	40.65	50.83	37.36
	Sparse Mean Field	31.19	27.84	31.51	35.37	38.68	48.39	35.50
Gradient-Modulated Occlusion Model - trained on all (for comparison)								
Disparity Error	Graph Cuts	10.61	19.2	5.98	20.95	7.15	5.53	12.78
	Sparse Mean Field	8.29	13.41	4.72	19.22	5.11	4.76	10.15
Occlusion Error	Graph Cuts	16.20	10.40	24.88	29.77	27.88	32.97	21.83
	Sparse Mean Field	10.47	8.10	19.43	23.04	21.10	27.31	16.43

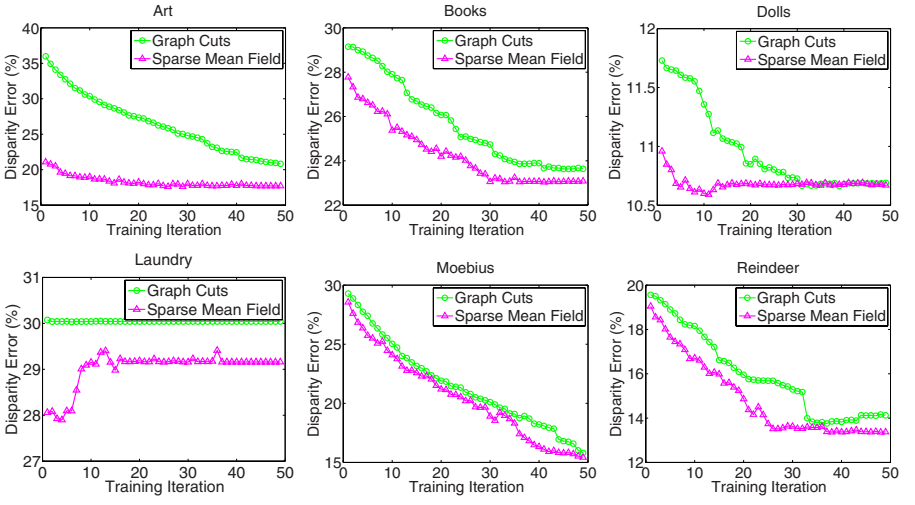


Fig. 6. Disparity error (each image held out in turn) using both graph cuts and mean field for learing the canonical CRF stereo model. The error before learning is omitted from the plots to better highlight performance differences.

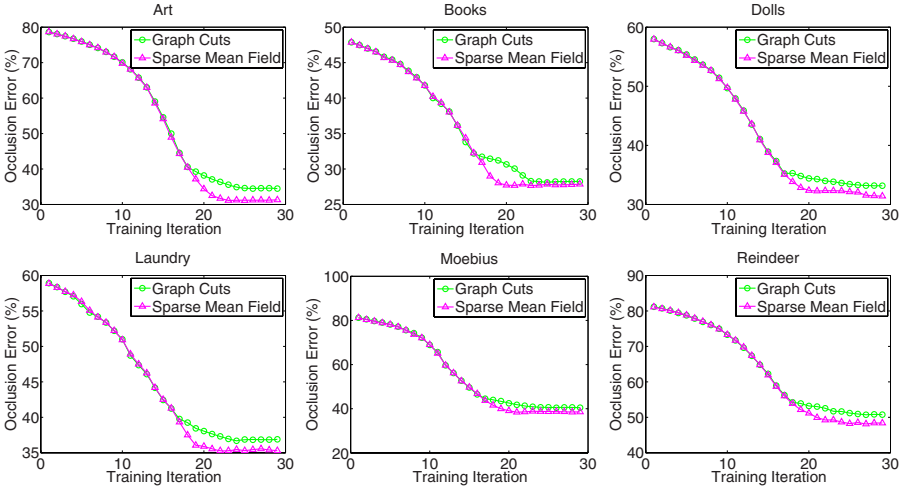


Fig. 7. Comparison of error predicting occluded pixel using graph cuts and sparse mean field for learning in the gradient-modulated occlusion model (5)

as training times are prohibitively long. For each experiment we leave out the image indicated and train on all the others listed. The disparity error is reduced by an average of $4.70 \pm 2.17\%$, and a paired sign test reveals the improvement is significant ($p < 0.05$).

We also test the error of our models' for occlusion predictions. We use the extended smoothness term (5) to handle the interactions between occluded states

and the local terms of (4). We show both leave-one-out training and test results as well as the result of training on all the data (as a reference point). Models trained using sparse mean field give more accurate occlusion predictions than the model trained using graph cuts. In the gradient-modulated occlusion model our leave-one-out experiments show that the error in predicting occluded pixels is reduced an average of $4.94 \pm 1.10\%$ and is also significant ($p < 0.05$).

Figure 6 shows that sparse mean field reduces the disparity error in the model more quickly than graph cuts during learning on many images. Even when the two methods approach each other as learning progresses, sparse mean field still converges at parameters providing lower errors on both disparity and occlusions (Figure 7).

5 Conclusions

In this paper, we have provided a framework for sparse variational message passing (SVMP). Calculating sparse updates to the approximating variational distribution can greatly reduce the time required for inference in models with large state spaces. For high resolution imagery this reduction in time can be essential for practical inference and learning scenarios. In addition, we have a variational bound on the cost of our approximation. Furthermore, compare to graph cuts, the resulting marginals of SVMP provide better parameter estimates when used for learning in a maximimum likelihood framework. Graph cuts is often the best at finding a low energy solution in a given model. However, for model learning, a distribution over configurations is required. In models where there is more uncertainty (as in the early stages of learning), we fond that sparse mean field provides a lower free energy than graph cuts. As such, our analysis indicates that SVMP can be used as an effective tool for approximating the distributions necessary for accurate learning. Sparse mean field can be seen as a method occupying a middle ground between producing point estimates and creating fuller approximate distribution.

Finally, one of the most important advantages of the sparse mean field technique is that one no longer has strong constraints on the forms of allowable potentials that are required for graph cuts. As such, we see sparse message passing methods a being widely applicable for models where the constraints on potentials imposed by graph cuts are too restrictive.

Acknowledgements: C.P. thanks Carestream and Kodak Research for helping support this research.

References

1. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusions using graph cuts. In: Proc. ICCV, pp. 508–515 (2001)
2. Sun, J., Li, Y., Kang, S.B., Shum, H.Y.: Symmetric stereo matching for occlusion handling. In: Proc. CVPR, pp. 399–406 (2005)

3. Yang, Q., Wang, L., Yang, R., Stewenius, H., Nister, D.: Stereo matching with color-weighted correlation, hierachical belief propagation and occlusion handling. In: Proc. CVPR (2006)
4. Yedidia, J., Freeman, W., Weiss, Y.: Understanding belief propagation and its generalizations. In: Exploring Artificial Intelligence in the New Millennium, pp. 236–239 (January 2003)
5. Jordan, M.I., Ghahramani, Z., Jaakkola, T., Saul, L.: Introduction to variational methods for graphical models. Machine Learning 37, 183–233 (1999)
6. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet allocation. J. of Machine Learning Research 3, 993–1022 (2003)
7. Frey, B.J., Jojic, N.: A comparison of algorithms for inference and learning in probabilistic graphical models. IEEE TPAMI 27(9) (September 2005)
8. Winn, J., Bishop, C.: Variational message passing. J. of Machine Learning Research 6, 661–694 (2005)
9. Andrieu, C., de Freitas, N., Doucet, A., Jordan, M.: An introduction to MCMC for machine learning. Machine Learning 50, 5–43 (2003)
10. Pal, C., Sutton, C., McCallum, A.: Sparse forward-backward using minimum divergence beams for fast training of conditional random fields. In: Proc. ICASSP, pp. 581–584 (2006)
11. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proc. ICML, pp. 282–289 (2001)
12. Kumar, S., Hebert, M.: Discriminative random fields. IJCV 68(2), 179–201 (2006)
13. Weinman, J., Hanson, A., McCallum, A.: Sign detection in natural images with conditional random fields. In: IEEE Intl. Workshop on Machine Learning for Signal Processing, pp. 549–558 (2004)
14. He, Z., Zemel, R.S., Carreira-Perpin, M.: Multiscale conditional random fields for image labeling. In: Proc. CVPR, pp. 695–702 (2004)
15. Scharstein, D., Pal, C.: Learning conditional random fields for stereo. In: Proc. CVPR (2007)
16. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. IEEE TPAMI 28, 1568–1583 (2006)
17. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE TPAMI 23(11), 1222–1239 (2001)
18. Birchfield, S., Tomasi, C.: A pixel dissimilarity measure that is insensitive to image sampling. IEEE TPAMI 20(4), 401–406 (1998)
19. Weinman, J., Pal, C., Scharstein, D.: Sparse message passing and efficiently learning random fields for stereo vision. Technical Report UM-CS-2007-054, U. Massachusetts Amherst (October 2007)

Recovering Light Directions and Camera Poses from a Single Sphere

Kwan-Yee K. Wong, Dirk Schnieders, and Shuda Li

Department of Computer Science,
The University of Hong Kong
Pokfulam Road, Hong Kong
{kykwong,sdirk,sdli}@cs.hku.hk

Abstract. This paper introduces a novel method for recovering both the light directions and camera poses from a single sphere. Traditional methods for estimating light directions using spheres either assume both the radius and center of the sphere being known precisely, or they depend on multiple calibrated views to recover these parameters. It will be shown in this paper that the light directions can be uniquely determined from the specular highlights observed in a single view of a sphere without knowing or recovering the exact radius and center of the sphere. Besides, if the sphere is being observed by multiple cameras, its images will uniquely define the translation vector of each camera from a common world origin centered at the sphere center. It will be shown that the relative rotations between the cameras can be recovered using two or more light directions estimated from each view. Closed form solutions for recovering the light directions and camera poses are presented, and experimental results on both synthetic and real data show the practicality of the proposed method.

1 Introduction

The recovery of light directions from images is an important problem in both computer vision and computer graphics. For instance, light directions can be used to infer shape from images using shape-from-shading algorithms. In image-based rendering and augmented reality, light information of a scene is exploited to render virtual/real objects into the scene seamlessly.

In the literature, there exists a relative large number of work dealing with light estimation. Early works [1,2,3,4] focused on estimating a single distant point light source. However, multiple light sources are often present in a natural environment, and the problem of estimating multiple illuminants is even more challenging. In [5], Yang and Yuille showed that shading information along the occluding boundaries imposes strong constraints on the light directions. They also noted that without extra information, a unique solution for more than four light sources cannot be computed from a single image under the Lambertian model. In [6], Zhang and Yang estimated multiple illuminants from a sphere of known physical size by identifying the critical points, which are often difficult

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 631–642, 2008.

to detect due to their sensitivity to noise. Takai et al. [7] proposed an approach based on two spheres to estimate near light sources. Wang and Samaras [8] combined shading and shadow cues to improve light estimation. The aforementioned methods are mostly based on the Lambertian model, and they all require the prior knowledge of the projections of some reference objects with known geometry to give the relationship between surface orientations and image intensities.

The specular reflection component of light is known to work in a very predictable manner, and it can be exploited for light estimation. A mirror ball was utilized in [9] to estimate the global illumination in a real world scene. Using such a mirror ball might, however, change the scene illumination due to its strong reflection property. Instead of using a purely specular sphere, spherical objects which exhibit both specular and diffuse reflections have been utilized in [10] and [11]. Powell et al. [10] used three spheres with known relative positions as a calibration object to triangulate the positions of light sources from specular highlights. Zhou and Kambhamettu [11] proposed an iterative method to recover the location and radius of a sphere from a pair of calibrated images, and used the recovered sphere to estimate the light directions from the specular highlights on the sphere. In [12], Li et al. combined multiple cues like shading, shadows and specular reflections to estimate illumination in a textured scene.

Similar to [11], this paper considers the problem of recovering multiple distance point light sources from a single sphere with unknown radius and location. Unlike [11] which requires multiple fully-calibrated views for recovering the radius and location of the sphere via an iterative method, it will be shown in this paper that light directions can be recovered directly from a scaled sphere estimated from a single view. Given multiple views of the sphere, a simple method is introduced to estimate the relative positions and orientations of the cameras using the recovered light directions. Hence, both the light directions and camera poses can be recovered using a single sphere. The proposed method will work under the assumption of a perspective camera with known intrinsics observing a sphere that reflects the specular component of multiple distant point light sources. As will be shown later, at least two distant point light sources are needed to uniquely determine the extrinsic parameters of the cameras.

The rest of this paper is organized as follows. Section 2 addresses the problem of sphere reconstruction from a single image. It is shown that with unknown radius, a one-parameter family of solutions will be obtained with all the sphere centers lying on the line joining the camera center and the true sphere center. Section 3 briefly reviews the standard technique for recovering light directions from the observed highlights of a sphere with known radius and location. It then proves that any sphere from the family of solutions recovered from a single image can be used to estimate the light directions. Section 4 presents a simple method for recovering the camera poses using the recovered sphere and light directions. Finally, experimental results on both synthetic and real data are given in Sect. 5, followed by conclusions in Sect. 6.

2 Scaled Reconstruction of Sphere

Consider a pinhole camera P viewing a sphere S. Without loss of generality, let the radius and center of the sphere be R and (X_c, Y_c, Z_c) respectively, and the camera coordinate system be coincide with the world coordinate system. The sphere S can be represented by a 4×4 symmetric matrix $\mathbf{Q}_s$ given by

$$\mathbf{Q}_s = \begin{bmatrix} \mathbf{I}_3 & -\mathbf{S}_c \\ -\mathbf{S}_c^T & (\mathbf{S}_c^T \mathbf{S}_c - R^2) \end{bmatrix}, \tag{1}$$

where $\mathbf{S}_c = [X_c \; Y_c \; Z_c]^T$ is the sphere center. Any 3D point X lying on S will satisfy the equation $\tilde{\mathbf{X}}^T \mathbf{Q}_s \tilde{\mathbf{X}} = 0$ where $\tilde{\mathbf{X}}$ represents its homogeneous coordinates. Suppose the 3×3 calibration matrix $\mathbf{K}$ of P is known, the projection matrix for P can be written as $\mathbf{P} = \mathbf{K}[\mathbf{I}\,\mathbf{0}]$. The image of S under P will be a conic C. This conic C can be represented by a 3×3 symmetric matrix $\mathbf{C}$, given by [13]

$$\begin{aligned} \mathbf{C} &= (\mathbf{P}\mathbf{Q}_s^* \mathbf{P}^T)^* \\ &= (\mathbf{K}\mathbf{K}^T - (\mathbf{K}\mathbf{S}_c/R)(\mathbf{K}\mathbf{S}_c/R)^T)^*, \end{aligned} \tag{2}$$

where $\mathbf{Q}^*$ denotes the dual to the quadric $\mathbf{Q}$, and is equal to $\mathbf{Q}^{-1}$ if $\mathbf{Q}$ is invertible. Any 2D point x lying on C will satisfy the equation $\tilde{\mathbf{x}}^T \mathbf{C} \tilde{\mathbf{x}} = 0$ where $\tilde{\mathbf{x}}$ represents its homogeneous coordinates.

The conic image C and the camera center will define a right circular cone which will be tangent to S, and the axis of this cone will pass through the sphere center $\mathbf{S}_c$ (see Fig. 1). If the radius R of the sphere S is known, $\mathbf{S}_c$ can be uniquely determined along this axis. In the next paragraph, a closed form solution for $\mathbf{S}_c$ will first be derived under a special case. The method for estimating $\mathbf{S}_c$ under the general case will then be discussed.

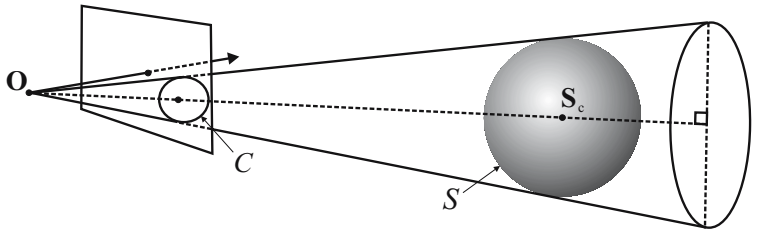


Fig. 1. The conic image C of the sphere S and the camera center will define a right circular cone. This cone is tangent to S and its axis passes through the sphere center $\mathbf{S}_c$.

Consider the special case where the sphere center lies along the positive Z-axis, and the camera calibration matrix is given by the identity matrix $\mathbf{I}_3$. Under this configuration, the sphere center will have coordinates $\mathbf{S}'_c = [0\; 0\; d]$. Note that d

is also the distance between the camera center and the sphere center. The image of the sphere can be obtained using (2), and is given by

$$\mathbf{C}' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{R^2}{R^2 - d^2} \end{bmatrix}. \tag{3}$$

Note that $\mathbf{C}'$ represents a circle with radius $r = \sqrt{-\frac{R^2}{R^2-d^2}}$. The center of $\mathbf{C}'$ is at the origin (i.e., $(0, 0)$), which is also the image of the sphere center. Given the radius r of $\mathbf{C}'$, the distance d between the camera center and the sphere center can be recovered as

$$d = R\frac{\sqrt{1+r^2}}{r}, \tag{4}$$

and the location of the sphere center follows.

Consider now the general case where the sphere center and the camera calibration matrix are given by $\mathbf{S}_\mathrm{c}$ and $\mathbf{K}$ respectively. Generally, the image of the sphere will no longer be a circle centered at the origin, but a conic $\mathbf{C}$ centered at an arbitrary point $\mathbf{x}_\mathrm{a}$. Note that $\mathbf{x}_\mathrm{a}$ is in general *not* the image of $\mathbf{S}_\mathrm{c}$. In order to recover $\mathbf{S}_\mathrm{c}$ from $\mathbf{C}$, the effect of $\mathbf{K}$ is first removed by normalizing the image using $\mathbf{K}^{-1}$. The conic $\mathbf{C}$ will be transformed to a conic $\hat{\mathbf{C}} = \mathbf{K}^\mathrm{T}\mathbf{C}\mathbf{K}$ in the normalized image. This conic $\hat{\mathbf{C}}$ can be diagonalized into

$$\hat{\mathbf{C}} = \mathbf{M}\mathbf{D}\mathbf{M}^\mathrm{T} = \mathbf{M}\begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & b \end{bmatrix}\mathbf{M}^\mathrm{T}, \tag{5}$$

where $\mathbf{M}$ is an orthogonal matrix whose columns are the eigenvectors of $\hat{\mathbf{C}}$, and $\mathbf{D}$ is a diagonal matrix consisting of the corresponding eigenvalues. The matrix $\mathbf{M}^\mathrm{T}$ defines a rotation that will transform $\hat{\mathbf{C}}$ to the circle $\mathbf{D}$ with radius $r = \sqrt{-\frac{b}{a}}$ centered at the origin. This transformation corresponds to rotating the camera about its center until its principle axis passes through the sphere center. This reduces the general case to the previously described special case, and the distance d between the camera center and the sphere center can be recovered in terms of r and R. Finally, the sphere center can be recovered as

$$\begin{aligned} \mathbf{S}_\mathrm{c} &= \mathbf{M}[0\ 0\ d]^\mathrm{T} \\ &= d\mathbf{m}_3, \end{aligned} \tag{6}$$

where $\mathbf{m}_3$ is the third column of $\mathbf{M}$.

3 Estimation of Light Directions

Suppose the center $\mathbf{S}_\mathrm{c}$ of a sphere with known radius R have been estimated using the method described in the previous section, it is then straightforward to recover the light directions from the observed highlights on the sphere. Standard

techniques begin by first constructing a ray from the camera center through a pixel corresponding to a highlight. The intersections of this ray with the sphere are then located to determine the point on the sphere giving rise to the highlight. By using the property that the angle of the incoming light must equal the angle of the outgoing light to the camera at a surface point with highlight, the light direction $\mathbf{L}$ can be recovered as

$$\mathbf{L} = (2\mathbf{N} \cdot \mathbf{V})\mathbf{N} - \mathbf{V}, \tag{7}$$

where $\mathbf{V} = \frac{\mathbf{X}-\mathbf{O}}{|\mathbf{X}-\mathbf{O}|}$ is the unit viewing vector, $\mathbf{N} = \frac{\mathbf{X}-\mathbf{S}_c}{|\mathbf{X}-\mathbf{S}_c|}$ is the unit surface normal vector at $\mathbf{X}$, $\mathbf{X}$ is a point with specular highlight on the sphere and $\mathbf{O}$ is the camera center.

Now suppose the radius R of the sphere is unknown, it has been shown in Sect. 2 that there exists a one-parameter family of solutions for the sphere center $\mathbf{S}_c$ which all lie on the straight line joining the camera center and the true sphere center. It will now be shown that the light direction recovered from an observed highlight using any of these scaled spheres will be identical. In other words, light directions can be recovered from the highlights observed in the image of a sphere without knowing its size and location.

Proposition 1. *Consider a ray casted from the camera center and the family of spheres with varying radius recovered from the conic image of a sphere. If this ray intersects any of these spheres, it will intersect all the spheres and the first point of intersection with each sphere will all have the same unit surface normal.*

Proof. Since the cone constructed from the camera center and the conic image of the sphere will be tangent to all the recovered spheres. Any ray lying within this cone will intersect all these spheres, whereas any ray lying outside this cone will intersect none of them.

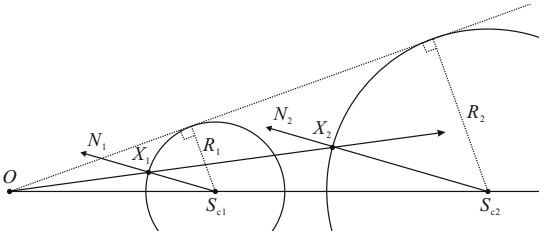


Fig. 2. The first intersection point of the ray with each sphere from the family of solutions will have the same unit surface normal

To prove that the intersection points have the same unit surface normal, it is sufficient to consider only the cross-section containing both the ray and the line defined by the sphere centers (see Fig. 2). Without loss of generality, consider a sphere S_1 from the family, and let its radius and center be r_1 and S_{c1} respectively. Suppose the ray intersect S_1 at X_1. The surface normal N_1

at X_1 is given by the vector $S_{c1}X_1$. Now consider a second sphere S_2 from the family, and let its radius and center be r_2 and S_{c2} respectively. A line being parallel to N_1 can be constructed from S_{c2}, and let the intersection point between this line and S_2 be X_2. By construction, the surface normal N_2 at X_2 will be parallel to N_1. Consider the two triangles $\triangle OS_{c1}X_1$ and $\triangle OS_{c2}X_2$. Obviously, $|X_1S_{c1}| : |X_2S_{c2}| = r_1 : r_2$. It follows from (4) that $|OS_{c1}| : |OS_{c2}| = r_1 : r_2$. Finally by construction, $\angle OS_{c1}X_1 = \angle OS_{c2}X_2$. Hence $\triangle OS_{c1}X_1$ and $\triangle OS_{c2}X_2$ are similar and $\angle S_{c1}OX_1 = \angle S_{c2}OX_2$. It follows that the ray will intersect S_2 at X_2 at which the surface normal N_2 is parallel to the surface normal N_1 at X_1. Since the two spheres being considered are chosen arbitrarily, the same argument applies to all spheres in the family, and the proof is completed. □

From (7), the light direction $\mathbf{L}$ only depends on the unit viewing vector $\mathbf{V}$ and the unit surface normal $\mathbf{N}$. The following corollary therefore follows immediately from Proposition 1:

Corollary 1. *The light direction estimated from an observed specular highlight in an image of a sphere will be independent of the radius used in recovering the location of the sphere center.*

4 Estimation of Camera Poses

Suppose two images of a sphere are captured at two distinct viewpoints. By applying the method described in Sect. 2 to each image independently, the sphere center can be recovered in each of the two camera-centered coordinate systems respectively. By assuming an arbitrary but fixed radius for the sphere in both views, it is possible to relate the two cameras in a common coordinate system. Without loss of generality, let the sphere center in the camera-centered coordinate system of the first view be $\mathbf{S}_{c1}$ and that of the second view be $\mathbf{S}_{c2}$ respectively. By considering a common world coordinate system centered at the sphere center, the projection matrices for the two views can be written as

$$\begin{aligned} \mathbf{P}_1 &= \mathbf{K}_1[\mathbf{I}\ \mathbf{S}_{c1}] \\ \mathbf{P}_2 &= \mathbf{K}_2[\mathbf{I}\ \mathbf{S}_{c2}]. \end{aligned} \tag{8}$$

Note that the above projection matrices are not unique. Due to the symmetry exhibited in the geometry of the sphere, an arbitrary rotation about the sphere center (i.e., the world origin) can be applied to the camera without changing the image of the sphere. This corresponds to rotating the camera around the sphere while keeping the cone constructed from the image tangent to the sphere. Hence, by choosing the first camera as a reference view, a more general form of the projection matrices for the two views is given by

$$\begin{aligned} \mathbf{P}_1 &= \mathbf{K}_1[\mathbf{I}\ \mathbf{S}_{c1}] \\ \mathbf{P}_2 &= \mathbf{K}_2[\mathbf{R}\ \mathbf{S}_{c2}], \end{aligned} \tag{9}$$

where $\mathbf{R}$ is a 3×3 rotation matrix with three degrees of freedom.

By assuming the light directions being fixed (globally) in both views, the highlights observed in the two images can be exploited to uniquely determine the relative rotation between the two cameras. Note that the location of the highlight on the sphere surface will depend on both the light direction and the viewpoint. Hence the locations of the highlights due to the same light direction will be different under two distinct viewpoints, and their projections on the two images do not provide a pair of point correspondence. Nonetheless, using the method described in Sect. 3, the light direction can be recovered in each of the two camera-centered coordinate systems. Without loss of generality, let the (unit) light direction in the camera-centered coordinate system of the first view be $\mathbf{L}_1$ and that of the second view be $\mathbf{L}_2$ respectively. Since these two directions are parallel in the common world coordinate system, the rotation matrix $\mathbf{R}$ relating the two cameras should bring $\mathbf{L}_1$ to $\mathbf{L}_2$, i.e.,

$$\mathbf{R}\mathbf{L}_1 = \mathbf{L}_2. \tag{10}$$

The above equation places two independent constraints on $\mathbf{R}$. Hence observing two highlights produced by two distinct light directions in two images will provide four constraints which are enough to determine $\mathbf{R}$ uniquely. Reader may refer to [14] for a robust method of determining a rotation from two of more pairs of directions using quaternion representation.

5 Experimental Results

The closed form solutions described in the previous sections for recovering light directions and camera poses have been implemented. Experiments on both synthetic and real data were carried out and the results are presented in the following sections.

5.1 Synthetic Data

The experimental setup consisted of a synthetic sphere being viewed by two synthetic cameras under two distinct directional lights. The conic images of the sphere were obtained analytically using (2). To locate the specular highlights in the images, the sphere was rendered by OpenGL with a Phong Shader using the viewing parameters of the two synthetic cameras (see Fig. 3). The specular highlights were then picked and matched manually, and a region growing technique was applied to extract the highlight regions from the images. The centroid of each region was then used as the highlight location for recovering the light directions.

In order to evaluate the robustness of the proposed method, uniformly distributed random noise was added to the conic images as well as to the locations of the specular highlights. To add noise to a conic, points were first sampled from the conic and perturbed in a radial direction from the conic center. A noisy conic was then obtained as a conic robustly fitted to these noisy points using a direct

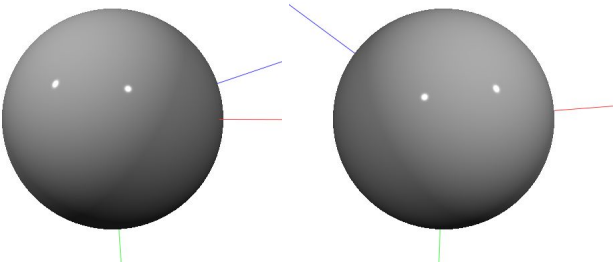


Fig. 3. Synthetic sphere rendered by OpenGL with a Phong Shader using the viewing parameters of the two synthetic cameras

least squares method [15]. For the location of a specular highlight, noise was added directly to its pixel coordinates.

Experiments on synthetic data with noise levels ranging from 0.1 to 3.0 pixels were carried out. For each noise level, 200 independent trials were conducted to estimate both the light directions and the camera poses from the noisy conics and highlights. Figure 4 shows a plot of the average angular error (in degrees) in the estimated light directions against the noise level (in pixels). It can be seen that the average error increases linearly with the noise level. For a noise level of 1.0 pixel, the average angular error in the estimated light directions is only about 0.5°. Figure 5 shows a plot of the average angular errors (in degrees) in the estimated rotations of the cameras against the noise level (in pixels). The rotation is represented here using a rotation axis, parameterized by the two

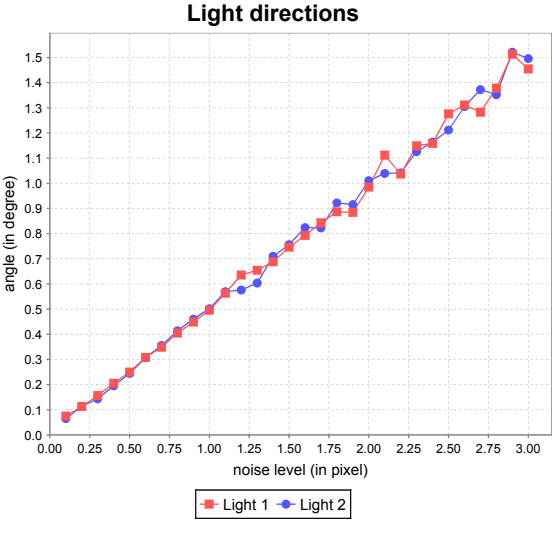


Fig. 4. A plot of the average angular error (in degrees) in the estimated light directions against the noise level (in pixels).

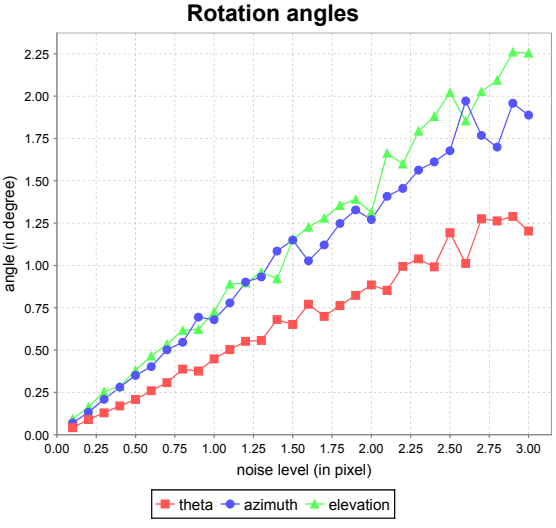


Fig. 5. A plot of the average angular errors (in degrees) in the estimated rotations of the cameras against the noise level (in pixels). The rotation is represented here using a rotation axis, parameterized by the two angles azimuth and elevation, and a rotation angle theta around the axis.

angles azimuth and elevation, and a rotation angle theta around the axis. It can be seen again that the average angular errors increase linearly with the noise level. This is expected as the computation of the rotation depends directly on the estimated light directions. For a noise level of 1.0 pixels, the average angular errors are less than 0.75° for the angles defining the rotation axis and less than 0.5° for the rotation angle around the axis.

5.2 Real Data

In the real data experiment, five light sources were placed approximately 3 meters away from a red snooker ball which has a radius of around 54mm. Seven images of the snooker ball were then taken at distinct viewpoints (see Fig. 6). The ground truth projection matrices for these seven views were obtained using a planar calibration pattern [16]. The intrinsic parameters of the camera were obtained by decomposing the ground truth projection matrices. Alternatively, the intrinsic parameters of the camera may also be recovered using existing techniques based on the conic images of the sphere [17,18]. Cubic B-spline snake was applied to extract the contours of the sphere in the images, and conics were then fitted to these contours using a direct least squares method [15]. Like in the synthetic experiment, the specular highlights were picked and matched manually from the images and a region growing technique was applied to extract the highlight regions from the images. The centroid of each region was then used as the highlight location for recovering the light directions. Since the camera poses were estimated up to an unknown scale, it is not very meaningful to directly

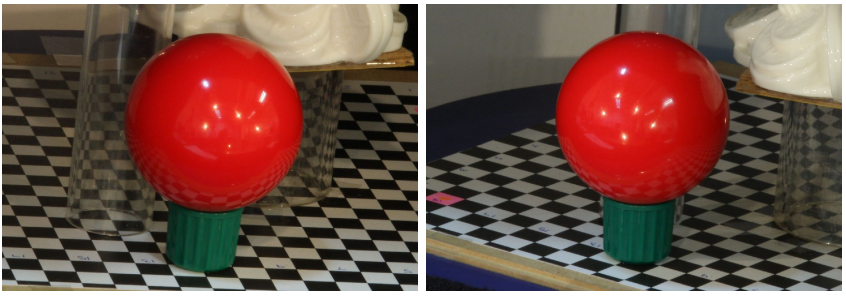


Fig. 6. Two images of a specular sphere taken under five distant point light sources

Table 1. Angular errors (in degrees) in the estimated rotations

view	theta	azimuth	elevation
1-2	0.2228	0.0178	0.1170
1-3	0.3264	0.0464	0.0089
1-4	0.1930	0.2250	0.1556
1-5	0.1587	0.1389	0.0197
1-6	0.0900	0.1079	0.0572
1-7	0.1020	0.1178	0.0110

compare the relative translations between the estimated cameras with those of the ground truth. Instead, the relative rotations between the estimated cameras were compared against those of the ground truth, and the angular errors in the estimated rotations are listed in Table. 1. Similar to the synthetic experiment, the rotation is represented here using a rotation axis, parameterized by the two angles azimuth and elevation, and a rotation angle theta around the axis. It can be seen from the table that the maximum angular error is about 0.33°, while most errors lie between 0.1° and 0.2°.

6 Conclusions

This paper addresses the problem of recovering both the light directions and camera poses from a single sphere. The two main contributions of this paper are

1. a closed form solution for recovering light directions from the specular highlights observed in a single image of a sphere with unknown size and location; and
2. a closed form solution for recovering the relative camera poses using the estimated sphere and light directions.

It is shown that given the intrinsic parameters of a camera, a scaled sphere can be reconstructed from its image. The translation direction of the sphere center from the camera center can be determined uniquely, but the distance between them will be scaled by the unknown radius of the sphere. It is then proved

that the light directions can be recovered independent of the radius chosen in locating the sphere. If the sphere is observed by multiple views, the sphere center recovered using a common fixed radius will fix the translations of the cameras from the sphere center. The relative rotations between the cameras can then be determined by aligning the relative light directions recovered in each view. As there exists closed form solutions for all the computation steps involved, the proposed method is extremely fast and efficient. Experiments on both synthetic and real images show promising results. With the proposed method, both the light directions and camera poses can be estimated simultaneously. This greatly eases the work of multiple views light estimation.

Acknowledgement

This project is supported by a grant from the Research Grants Council of the Hong Kong Special Administration Region, China, under Project HKU 7180/06E.

References

1. Horn, B.K.P., Brooks, M.J.: Shape and source from shading. In: International Joint Conference on Artificial Intelligence, pp. 932–936 (1985)
2. Pentland, A.P.: Finding the illuminant direction. Journal of Optical Soc. of America 72(4), 448–455 (1982)
3. Ikeuchi, K., Sato, K.: Determining reflectance properties of an object using range and brightness images. IEEE Trans. on Pattern Analysis and Machine Intelligence 13(11), 1139–1153 (1991)
4. Zheng, Q., Chellappa, R.: Estimation of illuminant direction, albedo, and shape from shading. IEEE Trans. on Pattern Analysis and Machine Intelligence 13(7), 680–702 (1991)
5. Yang, Y., Yuille, A.L.: Sources from shading. In: Proc. Conf. Computer Vision and Pattern Recognition, pp. 534–539 (1991)
6. Zhang, Y.F., Yang, Y.H.: Multiple illuminant direction detection with application to image synthesis. IEEE Trans. on Pattern Analysis and Machine Intelligence 23(8), 915–920 (2001)
7. Takai, T., Niinuma, K., Maki, A., Matsuyama, T.: Difference sphere: An approach to near light source estimation. In: Proc. Conf. Computer Vision and Pattern Recognition, vol. I, pp. 98–105 (2004)
8. Wang, Y., Samaras, D.: Estimation of multiple directional light sources for synthesis of augmented reality images. Graphical Models 65(4), 185–205 (2003)
9. Debevec, P.: Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with gobal illumination and high dynamic range photography. In: Proc. ACM SIGGRAPH, pp. 189–198 (1998)
10. Powell, M.W., Sarkar, S., Goldgof, D.: A simple strategy for calibrating the geometry of light sources 23(9), 1022–1027 (September 2001)
11. Zhou, W., Kambhamettu, C.: Estimation of illuminant direction and intensity of multiple light sources. In: Proc. 7th European Conf. on Computer Vision, vol. IV, pp. 206–220 (2002)
12. Li, Y.Z., Lin, S., Lu, H.Q., Shum, H.Y.: Multiple-cue illumination estimation in textured scenes. In: Proc. 9th Int. Conf. on Computer Vision, pp. 1366–1373 (2003)

13. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)
14. Horn, B.: Closed-form solution of absolute orientation using unit quaternions. Journal of Optical Soc. of America A 4(4), 629–642 (1987)
15. Fitzgibbon, A.W., Pilu, M., Fisher, R.B.: Direct least square fitting of ellipses. IEEE Trans. on Pattern Analysis and Machine Intelligence 21(5), 476–480 (1999)
16. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. on Pattern Analysis and Machine Intelligence 22(11), 1330–1334 (2000)
17. Agrawal, M., Davis, L.S.: Camera calibration using spheres: a semi-definite programming approach. In: Proc. 9th Int. Conf. on Computer Vision, pp. 782–789 (2003)
18. Zhang, H., Wong, K.Y.K., Zhang, G.: Camera calibration from images of spheres. IEEE Trans. on Pattern Analysis and Machine Intelligence 29(3), 499–503 (2007)

Tracking with Dynamic Hidden-State Shape Models

Zheng Wu[1], Margrit Betke[1], Jingbin Wang[2],
Vassilis Athitsos[3], and Stan Sclaroff[1]

[1] Computer Science Department, Boston University, Boston, MA, USA
[2] Google Inc.,Mountain View, CA, USA
[3] Computer Science and Engineering Department, University of Texas at Arlington, Arlington, Texas, USA

Abstract. Hidden State Shape Models (HSSMs) were previously proposed to represent and detect objects in images that exhibit not just deformation of their shape but also variation in their structure. In this paper, we introduce Dynamic Hidden-State Shape Models (DHSSMs) to track and recognize the non-rigid motion of such objects, for example, human hands. Our recursive Bayesian filtering method, called DP-Tracking, combines an exhaustive local search for a match between image features and model states with a dynamic programming approach to find a global registration between the model and the object in the image. Our contribution is a technique to exploit the hierarchical structure of the dynamic programming approach that on average considerably speeds up the search for matches. We also propose to embed an online learning approach into the tracking mechanism that updates the DHSSM dynamically. The learning approach ensures that the DHSSM accurately represents the tracked object and distinguishes any clutter potentially present in the image. Our experiments show that our method can recognize the digits of a hand while the fingers are being moved and curled to various degrees. The method is robust to various illumination conditions, the presence of clutter, occlusions, and some types of self-occlusions. The experiments demonstrate a significant improvement in both efficiency and accuracy of recognition compared to the non-recursive way of frame-by-frame detection.

1 Introduction

The radar literature describes mature algorithms for tracking targets that estimate the state of the targets within a dynamic system using recursive Bayesian filters [1]. The computer vision community has extended these algorithms to track objects in images – a much more difficult problem because the appearance of deformable and articulated objects in images can vary enormously. Our work moves beyond the limit of what has been accomplished so far by formulating and solving the task of tracking objects that have a variable shape structure (in addition to being rigid or deformable, and/or articulated). We present a model-based, recursive Bayesian estimation algorithm that tracks such objects in images with

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 643–656, 2008.

heavy clutter and simultaneously recognizes the various components of the object. Since the object has a variable structure, our method does not know a priori if any of these components appear or disappear in the video and what shape they will exhibit. An example of an object with variable structure is a hand, whose components, the fingers, may or may not appear in an image and may be curled to various degrees, resulting in self-occlusion (Fig. 1(a)). Our tracking method uniquely identifies and outlines each finger in the video, regardless of the degree of curling motion. Moreover, it concurrently handles dense clutter and illumination changes, and recovers automatically from occlusions by other objects. Our contributions are:

- We introduce dynamic hidden-state shape models (DHSSMs) to model classes of moving objects of variable structure (Fig. 1). DHSSMs are a generalization of (static) hidden-state shape models (HSSMs) [2,3], where the model description will be updated through time.
- We propose a dynamic programming (DP) approach, called DP-Tracking, to find an optimal registration between the model states of a DHSSM and the features extracted from each video frame. DP-Tracking speeds up the model registration by two orders of magnitude compared to Wang et al.'s approach [3] and tracks hands in near real time. We achieve this by introducing a grouping technique that takes advantage of the spatial coherence of features in the DP table during tracking.
- We embed an online learning approach into the tracking mechanism that captures appearance changes through time and produces tracking results that are significantly more accurate than the results obtained with DP-Tracking without online learning or Wang et al.'s approach [3].

Our data contains a large number of candidate features (about 3,000 edge points per frame). We need to process both the contours of the object to be tracked and background objects (clutter) or occluding foreground objects (also clutter). Our method tracks each feature at position p on the contour of the moving object by assigning it to a feature in the next frame that is near to p. This straightforward recursive Bayesian tracking approach is particularly suited for such data, more so than the Kalman, kernel-based, or particle filters [1,4,5]. It is challenging to use the latter in real time processing, since the number of particles required would increase exponentially with the dimension of the state space. The hand DHSSM, for example, has 20 states including position, orientation, scale and feature duration (i.e., finger's length). Nevertheless, our framework for tracking objects with DHSSMs in principle allows adoption of any Bayesian algorithm, and others may explore this in the future.

Related Work. Tracking algorithms typically assume smoothness: the state or the appearance of an object component does not change much from one image frame to the next. This assumption helps to overcome problems with tracking when the interpretation of image data is ambiguous. Successful systems have used graphical models for modeling non-rigid 3D objects [6,7]. These systems, however, require a manual initialization of the object to be tracked (our

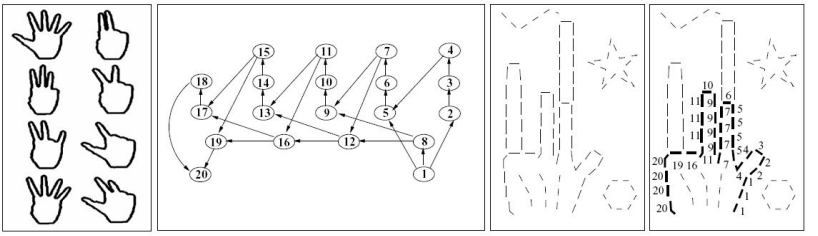


Fig. 1. Recognizing hands in images using HSSMs. (a) Hand contours with variable shape structure. (b) State transition diagram of the HSSM "hand." State s_1 models the right boundary of the palm, s_{20} the left. States s_2,...,s_4 model the thumb; in particular, state s_2 the right side of the thumb; state s_3 the thumb tip, and state s_4 the left side of the thumb. States s_5,...,s_7, s_9,...,s_{11}, s_{13},...,s_{15}, and s_{17}, s_{18} respectively model the contour of the other four fingers. States $s_8, s_{12}, s_{16}, s_{19}$ model the occluding boundary when the fingers are totally bent. (c) An edge image with segments of the hand contour and clutter. (d) Registration of hand model states to contour segments, resulting in the recognition of the hand structure. Figure courtesy of Wang et al. [3].

system does not), and there is no guarantee that the systems can recover when the previous frame is interpreted falsely. An alternative way to overcome the problem of losing track is to treat tracking as the problem of detecting the object in each frame [8,9,10], also known as *tracking by detection* [11]. Thus, poor tracking in one frame will not affect any subsequent results, and the problem of drifting is also prevented [12]. Our DHSSM-based method extends the tracking-by-detection idea by adding "gating" [1], and is able to recover from tracking errors as we show in the experiments.

A popular approach for tracking-by-detection is template matching (e.g., [10]). To deal with object deformation, multiple templates can be organized in a tree structure so that exhaustive search for the best match can be replaced by hierarchical matching [10]. Although template matching is efficient, it may be difficult to use it to identify clutter or deal with large within-class variations in object shape. Recognition approaches that exploit contour-based information recently received considerable attention in the object detection literature (e.g., [13,14,15]). These algorithms either use static boundary templates or codebooks of appearance parts. Note the difference to our contour-based work: these representations model standard object configurations and do not explicitly account for the variable structure of the objects (e.g., fingers that are totally extended, partially bent, or completely hidden, as in Fig. 1). Another classic work of tracking 2D silhouette deals with the problem of discontinuous shape changes by explicitly modelling *wormholes* in shape space [16], but the contour extraction is not designed to handle cluttered images.

It is desirable to use rich models that can capture a large range of possible object variations and efficient methods that can register the image data to such models in tracking scenarios. In this paper, we propose such models, DHSSMs,

and apply them with an efficient DP search algorithm to perform detection, recognition, and tracking of human hand in heavily cluttered images.

2 Dynamic Hidden-State Shape Models

We extend (static) Hidden-State Shape Models (HSSMs) [3], which are based on HMMs [17], to Dynamic Hidden-State Shape Models (DHSSMs) to include the index of the current frame, denoted by superscript t. A DHSSM is defined by

- a set $\mathbb{S} = \{s_1, \ldots, s_M\}$ of states modeling object components.
- a subset $\mathbb{E}$ of $\mathbb{S}$ that defines legal end states,
- the probability $\pi^{(t)}(s_i)$ that state s_i is the initial state at time t,
- the *state transition function* $A^{(t)}(s_i, s_j) = p^{(t)}(s_j|s_i)$ that represents the probability of transiting from state s_i to state s_j at time t,
- the *state observation function* $B^{(t)}(f_u, s_i) = p^{(t)}(f_u|s_i)$ that represents the probability of observing feature f_u in state s_i at time t,
- the *feature transition function* $\tau^{(t)}(f_v, f_u, s_j, s_i) = p^{(t)}(f_v|f_u, s_j, s_i)$ that represents the probability of observing feature f_v in state s_j given some other feature f_u was previously observed in state s_i at time t, where s_i could be equal to s_j if both f_u and f_v belong to the same object part, and
- the *state duration function* $D^{(t)}(\ell, s_i, \omega) = p^{(t)}(\ell|s_i, \omega)$ that represents the probability of continuously observing ℓ features in state s_i at time t, given the prior object scale ω.

To recognize the structure of an object modeled by a DHSSM in a frame $I^{(t)}$, we use the approach by Wang et al. [3] to extract K features $\mathbb{O}^+$ from the image, recognize the L features $\mathbb{O} \subseteq \mathbb{O}^+$ that represent the object contour, and separate them from the $K-L$ features $\mathbb{O}_c \subset \mathbb{O}^+$ that represent clutter. This is achieved by finding an *ordered* match between the object features $\mathbb{O} = (o_1, ..., o_L)$ and their corresponding DHSSM states and a match between the features $\mathbb{O}_c = (o_{L+1}, ..., o_K)$ that are not on the object contour and a special state $q_c \notin \mathbb{S}$ that is introduced to represent clutter. By $(o_j : q_i)$, we denote the match between an observed feature $o_j \in \mathbb{O}^+$ and a state $q_i \in \mathbb{Q}^+ = \mathbb{S} \cup \{q_c\}$. Since several features on the contour may be assigned to the same state, we use the notation $(O_j^{(d)} : q_i)$ to describe the ordered match between d observed features $O_j^{(d)} = (o_j, ..., o_{j+d})$, forming a single contour segment (e.g., side of a finger), and state q_i. A registration R of n contour segments to a sequence of n states is then

$$R^{(t)}_{\mathbb{O},\mathbb{Q}} = [(\mathrm{O}_1^{(\mathrm{d}_1)} : \mathrm{q}_1),\ (\mathrm{O}_{\mathrm{d}_1+1}^{(\mathrm{d}_2)} : \mathrm{q}_2),\ \ldots,\ (\mathrm{O}_{\mathrm{L}-\mathrm{d}_\mathrm{n}+1}^{(\mathrm{d}_\mathrm{n})} : \mathrm{q}_\mathrm{n})]^{(t)},$$

where $\mathrm{O}_j, \mathrm{q_i}, \mathrm{d_i}, \mathrm{n}$, and L are considered random variables. To recognize the object in the t-th video frame, the registration algorithm must find values for these random variables that maximize the joint probability $p(\mathbb{O}^+, \mathbb{Q}^+)$ of observations and states. By extending the derivations of Wang et al. [3] in a straightforward manner to include the index t, we can show this to be equivalent to minimizing the registration cost function

$$C^{(t)}(R^{(t)}_{\mathbb{O},\mathbb{Q}}) = -\ln \pi^{(t)}(\mathrm{q}_1) - \sum_{i=1}^{\mathrm{n}} \{\ln A^{(t)}(\mathrm{q}_{i-1}, \mathrm{q}_i) + \ln D^{(t)}(\mathrm{d}_i, \mathrm{q}_i, \omega) - \xi(\mathrm{d}_i) + \sum_{j=\zeta(i)+1}^{\zeta(i)+\mathrm{d}_i} [\ln \tfrac{B^{(t)}(\mathrm{o}_j, \mathrm{q}_i)}{B^{(t)}(\mathrm{o}_j, q_c)} + \ln \tau^{(t)}(\mathrm{o}_j, \mathrm{o}_{j-1}, \mathrm{q}_i, \mathrm{q}_{i'})]\}, \quad (1)$$

where $A^{(t)}, B^{(t)}, D^{(t)}$ and $\tau^{(t)}$ were introduced above, $\xi(\mathrm{d}_i) = \mathrm{d}_i \ln p(q_c)$, $\zeta(i) = \sum_{k=1}^{i-1} \mathrm{d_k}$, and $i' = i - 1$ when $j = \zeta(i) + 1$ and $i' = i$ otherwise. Recognition of the object in the presence of clutter is achieved by finding the globally optimal registration $R^{(t)}_{\mathrm{opt}} = \arg\min C^{(t)}(R^{(t)}_{\mathbb{O},\mathbb{Q}})$. More details about HSSMs can be found in Wang et al. [3].

For (static) HSSMs, a DP algorithm for minimizing the cost function C was proposed [3], which has the computational complexity of $O(MC_sK^2\ell_{\max})$, where M and K are the number of states and features, C_s is the average number of legal state transitions out of each state, and $\ell_{\max}$ is the maximum number of features that can be observed in a state. The `C++` implementation of this algorithm processed a 160x120 image in about 25 minutes (AMD Opteron 2.0 GHz processor) to determine the size and pose of a single object of variable structure. This included an exhaustive search among eight possible object orientations and no feature pruning (K is up to 3000). Thus, a frame-by-frame application of Wang et al.'s HSSM-based algorithm [3] to track objects in long video sequences is computationally inefficient. Here we show how DHSSMs and a new DP-based registration algorithm can be combined into a fast and robust object tracking system. The essential steps of our DHSSM-based method comprise hierarchical dynamic programming and online model updates.

2.1 Specification of a DHSSM for the Hand

We define the hand DHSSM similar to the hand HSSM introduced by Wang et al. [3] (Fig. 1). An image feature $f \in \mathbb{O}^+$ is a local image patch surrounding an edge pixel, measured by its appearance ϕ and location y, i.e., $f = (\phi, y)$. The appearance ϕ is specified by the color distribution ϕ_χ of the 5×5 pixels patch (a vector that stores weighted average rgb values for each of the two half patches separated along the edge direction in the patch center) and the intensity gradient orientation ϕ_g at the patch center (a scalar that ranges from 0 to 2π). For each object state $s \in \mathbb{S}$, we model the color distribution $s^{(t)}_\chi$ of an object boundary patch and the gradient $s^{(t)}_g$ at the center of the patch at t.

The state transition function $A^{(t)}$ can be simplified as a uniform distribution. We define the object/clutter observation likelihood ratio as

$$\frac{B^{(t)}(f,s)}{B^{(t)}(f,c)} = \frac{p^{(t)}(\mathrm{o} = f \mid \mathrm{q} = s)}{p^{(t)}(\mathrm{o} = f \mid \mathrm{q_c} = c)} \approx e^{-\gamma\, h^{(t)}(\phi_\chi)} p(\phi_g | s^{(t)}_g), \quad (2)$$

where $p(\phi_g | s^{(t)}_g)$ is a Gaussian distribution with mean $s^{(t)}_g$ and variance σ^2_g. Function $h^{(t)}$, for given input ϕ_χ, outputs a decision value, which is approximated by a two-class Support Vector Machine (SVM). The scalar factor γ is determined experimentally. The feature transition probability

$$\tau^{(t)}(f, f', s, s') = e^{-\alpha(\|y'-y\|)} e^{-\beta|\Delta(\phi_g, \phi_{g'}) - \Delta(s^{(t)}_g, s^{(t)}_{g'})|} \quad (3)$$

is an exponential distribution, where $\|y - y'\|$ represents the Euclidean distance between the centers of the two patches f and $f,'$ and $\Delta(\phi, \phi')$ represents the angle in radians between orientations ϕ and ϕ'. The weighting scalars α and β can be learned from the training data by maximum likelihood estimation. The state duration probability $D^{(t)}$ is defined as the Gaussian model

$$D^{(t)}(\ell, s, \omega) = p(\ell \mid \mu^{(t)}(\omega), \sigma^{(t)}(\omega))\, p(\mu^{(t)}(\omega)|s) \tag{4}$$

where ω is an input parameter to specify the reference scale, $p(\ell \mid \mu^{(t)}(\omega), \sigma^{(t)}(\omega))$ is a normal distribution with the mean $\mu^{(t)}(\omega)$ and covariance $\sigma^{(t)}(\omega)$, and $p(\mu^{(t)}(\omega) \mid s)$ is the conditional prior for the Gaussian distribution.

3 Tracking with DHSSMs

An overview of our tracking system for the application of hand and finger tracking is given in Fig. 2. The system contains the hand DHSSM that is updated to capture appearance changes through time, for example, due to the occluding

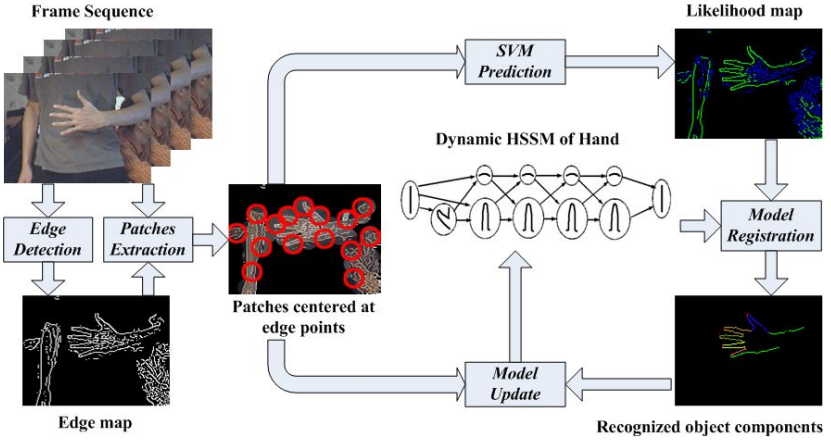


Fig. 2. Overview of tracking system. Each input image is first processed to extract features (edges) and feature patches (image regions centered around edges), pruned by a learned skin color model. The two-class SVM determines which features are likely to belong to the object contour (foreground) and which not (background clutter or occluding foreground clutter). Hand and finger detection is achieved by finding a globally optimal registration between DHSSM states and likely features. The locations of recognized object components (finger tips, sides, palm, etc.) are the frame-by-frame output of the tracking system and used to update the DHSSM online. After each frame, the probability densities of each state of the DHSSM are updated by feeding back the estimated orientation of the hand, and the relative orientation and amount of curling of each finger (the latter via updating the state duration variable). From time to time, a new collection of object and clutter patches are sampled to train a new SVM classifier that better models the current imaging scenario.

motion of curling fingers or some other objects (short-term update) or changes in illumination and background (long-term update). Pseudo-code of our DP-TRACKING algorithm is given in Sec. 3.2.

3.1 Updating the DHSSM During Tracking

We first explain how to update DHSSM in tracking. The temporal information is used to update the DHSSM in the short term, i.e., after processing each frame, and the SVM in the long term, i.e., after processing a number of frames.

The appearance description $s_g^{(t)}, \mu^{(t)}(\omega), \sigma^{(t)}(\omega)$ of the DHSSM is updated for each state s by sampling from its posterior distribution. For each object state $\mathrm{q} = s \in \mathbb{S}$ of the DHSSM, the posterior distribution of the image gradient s_g at the center of an object boundary patch is estimated from the optimal registration in the previous frame $t-1$, denoted as $R_{\text{opt}}^{(t-1)}$:

$$p(s_g^{(t)}|R_{\text{opt}}^{(t-1)}) \sim \mathcal{N}(\mu(\phi_g^{(t-1)}), \sigma^2(\phi_g^{(t-1)})) \tag{5}$$

where $\mu(\phi_g^{(t-1)})$ is the mean gradient direction of feature set $\{o_j^{(t-1)}, \ldots, o_{j+d}^{(t-1)}\}$ that mapped onto state s and $\sigma^2(\phi_g^{(t-1)})$ the variance. This step captures the change of orientation of each state.

The posterior distribution of the duration mean $\mu(\omega)$ for each state s is estimated from the optimal registration during the past N frames,

$$p(\mu^{(t)}(\omega)|\underline{R_{\text{opt}}^{(t-1)}}) \sim \mathcal{N}(\mu(\underline{d}^{(t-1)}), \sigma^2(\underline{d}^{(t-1)})) \tag{6}$$

where $\underline{R_{\text{opt}}^{(t-1)}}$ is the history of the past N optimal registrations, $\mu(\underline{d}^{(t-1)})$ is the mean number of contour points mapped onto state s during the past N optimal registrations, and $\sigma^2(\underline{d}^{(t-1)})$ is the variance. We set $\sigma^{(t)}(\omega)$ to be $\sigma(\underline{d}^{(t-1)})$ for simplicity.

The long-term update of the tracking system focuses on the color distribution s_χ of a boundary patch for each object state s. As mentioned in Sec. 2.1, we approximate this log-likelihood ratio by the output of a two-class SVM. However, it is challenging to build a color model that is robust and sufficiently discriminative under various illumination conditions and background scenes [18]; defining a robust skin color model remains a research topic by itself. It is therefore natural to perform ONLINE LEARNING, where the training set comprises of "positive patches" sampled along the detected hand boundary, and "negative patches," sampled from clutter. Note we only train a new classifier every few dozens of frames, so that training time is still acceptable for a classifier like an SVM. We adopted two conservative strategies when we rebuild the classifier:

(1) Avoid sampling negative patches located close to the object boundary, where they could be misclassified due to shadows, occlusion, noise, etc.

(2) Add a validation step before training samples are collected to ensure that the samples are labelled correctly. To validate our registration result in the current frame, we assume that the registration result of the previous frame is trustworthy and create a hand contour template from it. We then perform chamfer

matching between this contour template and the contour found in the current frame. A matching cost above a certain threshold is considered suspicious, and it is then decided to suspend all model updates.

3.2 Exploiting the Hierarchical Structure in DP

Our hand DHSSM resembles a left-to-right HMM (no state-transition loops are allowed) and we can build a 2D dynamic programming table to organize the matching process (Note the general DHSSM can have state-transition loops). The table has two axes. The state axis represents an ordered sequence of model states and each state contains multiple rows showing the state duration; the feature axis represents an *unordered* sequence of input features. We do not have ordered sequences in both axes as for Dynamic Time Warping [19].

To ease the explanation, we assume the state transition is a single Markov chain. The DP process can be seen as finding the shortest path between s_1 and s_M (e.g., right and left side of the palm) in a directed graph. Each node in the graph represents a matching pair $(f_j : s_{i,l})$, where feature f_j is observed in state s_i with duration l. Each edge that links two matching pairs $(f_j : s_{i,l_1})$ and $(f_n : s_{m,l_2})$ represents a possible jump from the observation f_j in s_i with duration l_1 to the next observation f_n in s_m with duration l_2. The weight of an edge is defined by the matching cost. To avoid the case of a loop, i.e., explaining one feature by more than one state, we restrict the selection of a set of features in the next jump, excluding those features that have already been chosen to belong to the shortest path. A DP algorithm to find the shortest path in this table generally has to consider all possible paths through the graph and therefore requires $O(MK^2\ell_{\max})$ operations, where M and K are the number of states and features, and $\ell_{\max}$ is the length of longest segment that can be observed. Since we use patches around edge points to represent image features, K typically is quite large, even in moderate sized images (up to 3,000 in the 160x120 video frames in our experiments, see Sec. 4). Gating [1] in the next frame will reduce the number of candidate features. In addition, we can exploit information from the spatial arrangement of the edge points in the local search neighborhood to help speed up the dynamic programming approach.

A benefit from tracking is that we can group neighboring edge points in the current frame based on the registration result in the previous frame. This yields groups of unordered input features. For example, in Fig. 3, the locations of feature points possibly matching onto states s_1 and s_2 are constrained by their respective local neighborhoods or gates (light and dark gray regions). A state transition can only happen in the intersection of the light and dark gray regions. when constructing the DP table, all the feature points are now sorted into groups where the group order corresponds to the state order. During the matching process, only feature transitions within the same group will be considered. Any valid shortest path must pass through all the intersection regions. In other words, within each group, we constrain the valid start and end points to be located in intersection regions.

DHSSM-based Tracking Algorithm:

Given Initial DHSSM description: $\{\mathbb{S}, \mathbb{E}, \pi^{(1)}, A^{(1)}, B^{(1)}, \tau^{(1)}, D^{(1)}\}$, skin color model, SVM classifier, and optimal registration in first frame $R_{opt}^{(1)}$ computed by [3].
For frame $t = 2...N$:

1. **Data Preparation**
   - Extract and prune features $\mathbb{O}^+$ from frame t by gating and skin color model;
   - Grouping based on $R_{opt}^{(t-1)}$;
   - Construct DP table.
2. **Model Registration**
   - For each group of features, running DP to find k candidate shortest paths;
   - For each group, detect occlusion by comparing histograms of gradients along candidate paths with corresponding segment in $R_{opt}^{(t-1)}$. If occlusion exists, label such groups invalid;
   - Run DP by linking one candidate valid path from each group to find the optimal shortest path $R_{opt}^{(t)}$.
3. **DHSSM Update**
   - Short-term update:
     Sample $s_g^{(t)}, \mu^{(t)}(\omega)$ according to Eqs. 5 and 6 and set $\sigma^{(t)}(\omega)$ to be $\sigma^2(\underline{d}^{(t-1)})$;
   - Long-term update: Collect $\mathbb{Q}^+, \mathbb{O}^+$ from $\underline{R_{\text{opt}}^{(t-1)}}$ to train a new SVM.

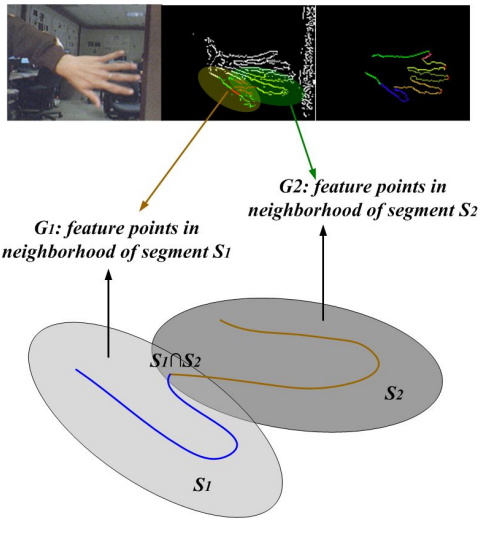


Fig. 3. Feature candidates mapped onto state s_1 are constrained by the light gray search region (gate), while feature candidates mapped onto state s_2 are constrained by the dark gray region. State transition can only happen in the intersection $s_1 \cap s_2$ of the two regions.

If there are G groups, the average number of feature points within each group is $\frac{K}{G}$, and we need $O(\frac{MK^2 l_{max}}{G^2})$ operations to find the optimal registration. A careful design of the DHSSM is critical to achieve the proposed reduction in

computation. If more than two group regions overlap each other, there is no guarantee that the optimal path will not contain feature transitions between non-adjacent groups. One way to deal with multiple overlapping regions is to insert duplicated features when grouping, which causes the input feature size to increase. Usually a proper choice of the size of the gate or neighborhood and an appropriate design of the states that correspond to large segments of the object boundary can alleviate the problem of redundant representation of feature points that fall into multiple overlapping regions.

Another benefit from the proposed grouping is that dynamic programming can be applied within each group independently. This step produces a number of candidate paths that correspond to boundary segments in the image. Then DP can be applied one more time by selecting one segment from each group and linking them together to form the final shortest path corresponding to the whole object boundary. The hierarchical DP algorithm allows the freedom to model the deformation within each segment locally and enforces spatial coherence between segments globally.

Moreover, it provides the opportunity to detect occlusions and help solve the problem of partial matches. While running DP within each group, our method compares histograms of gradients along each of those candidate paths with that of corresponding segments detected in previous frame. Intuitively, when occlusion occurs in some group, the candidate paths are noisy shapes so that the gradient histogram is quite different from that in the previous frame. If our method determines that occlusion occurs in some groups, candidate paths within these groups are excluded from participating in the second DP calculation. Then an incomplete hand contour can still be fitted well to the DHSSM model.

4 Experiments and Results

We implemented our DHSSM-based method in `C++` and tested it on an AMD Opteron 2.0 GHz processor. The datasets are challenging: 1) the background contains dense clutter so that the edges of the hand boundary and edges of clutter objects are difficult to distinguish; 2) the background is affected by illumination changes and other moving objects; 3) some scenes contain faces which means that skin-color-based hand detection must overcome ambiguities; 4) the shape structure of the hands in the videos changes due to the curling motion of the fingers, 5) the hand might be occluded so that there is no guarantee that a complete hand boundary exists in the scene. We assert that our datasets are sufficiently representative to demonstrate the performance of our system in tracking hands and fingers in real environments.

We used two quantitative measures of detection accuracy: (1) **Hand localization**: the algorithm correctly located the hand's position and orientation. This can be considered as the minimum requirement for all tracking systems. (2) **Finger identification**: the algorithm not only correctly located the hand's position and orientation, but also identified the state of each finger. We required that every visible finger was registered correctly to the respective states

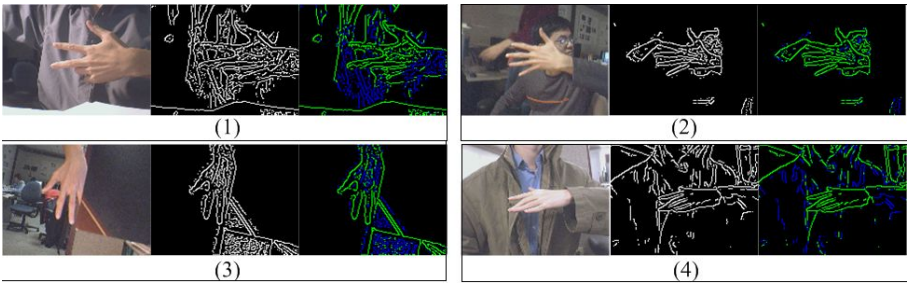


Fig. 4. Four 160x120 input images (left), edge maps after pruning with the skin color model (middle), and likelihood ratio maps predicted by the SVM classifier (right) with likely hand contour edges in green and clutter in blue. Samples include curled fingers (1), skin-color clutter (2), occlusion (3), and out-of-plane rotation (4).

of the hand DHSSM. Such registration information would be valuable for solving advanced image understanding tasks, such as sign language recognition. Four datasets, collected in a laboratory environment, were used in our experiments:

(1) Data with large motion of the hand and fingers (260 frames): the hand appeared in the video in different positions, orientations, and scales because of its translation, rotation, and movements towards and away from the camera. Deformation due to different degrees of curling of fingers was included.
(2) Data with dense clutter (510 frames): the background contained other moving objects (walking people) and skin-color patches (faces).
(3) Data with illumination changes (182 frames): the scene and object color changed significantly due to newly-appearing light sources.
(4) Data with occlusions (167 frames): a part of the hand was occluded by the motion of a background object so that partial matching of the hand was required.

Our DHSSM-based method correctly localized the hand in almost all tested situations that involved a large amount of motion and clutter (98% and 92% among all frames in test dataset 1 and 2, respectively). The shape of each of the five fingers was also detected in these situations (89% and 85%, respectively). Illumination changes and occlusion reduced the recognition percentages by about 20 percentage points (Table 1). Online learning contributes to increasing the accuracy of finger identification (85%) by 10 percentage points when the skin color varies due to changing lighting conditions (Table 1, Fig. 5, middle). Our method was particularly useful in interpreting the hand in the presence of occluding objects whose appearance strongly interfered with feature detection, e.g., the corner of the notebook in Fig. 5, right.

Our DHSSM-based method processes each video frame in near real time. The running time is proportional to the number of edge points that must be processed, e.g., for datasets with less than 2,000 edge points on average per frame, it was 1 s and with 3,000 edge points 1.5 s (Table 1).

To compare our method to a previously published method, we reimplemented Wang et al.'s [3] HSSM-based method. For a fair comparison of both methods,

Table 1. Comparison of our DHSSM-based method with DP-Tracking (DPT) and with and without Online Learning (OL), and Wang et al.'s [3] method (HSSM)

Dataset	Large Motion		Dense Clutter		Partial Occlusion		Illumination Change	
Avg. # of features	1,200		1,800		3,000		3,000	
Method	HSSM	DHSSM +DPT	HSSM	DHSSM +DPT	HSSM	DHSSM +DPT	DHSSM +DPT	DHSSM +DPT +OL
Hand Localization	96%	98%	92%	92%	55%	75%	83%	83%
Finger Identification	75%	89%	79%	85%	40%	65%	58%	68%
Avg. time/frame	160 s	1 s	160 s	1 s	250 s	1.5 s	1.5 s	2 s

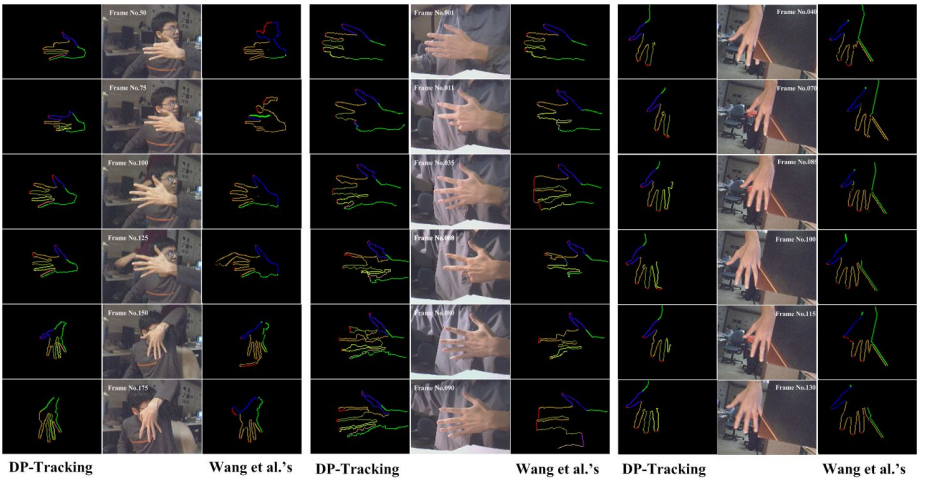


Fig. 5. Tracking results for images with clutter (left) due to a face, a chair, and a person moving in the background, and illumination changes (middle), and occlusions (right). The colors of the boundary pixels indicate which part of the boundary was detected as the finger tips (red), thumb (blue), index finger (orange), middle finger (olive), ring finger (yellow), little finger (pink), and outer boundary of the little finger and wrist (green). DP-Tracking with online learning produced the best recognition results.

we used the same size of the local search window for each feature point (40×40 pixels). Using Wang et al.'s method, we registered the static HSSM of the hand to the image data in each frame separately. Only the global orientation of the hand was passed from frame to frame. The processing time of our method compared to Wang et al.'s method [3] was more than two orders of magnitude shorter (Table 1). The improvement in localizing the hand was up to 20 percentage points and in identifying the fingers up to 25 percentage points (Table 1 and Fig. 5). The system has the ability to recover from tracking errors, which can occur due to severe occlusions or self-occlusions.

5 Discussion and Conclusion

We developed a near-real-time system that not only tracks a fast-moving hand in a cluttered environment but, at the same time, recognizes and labels the state of each moving finger. These tasks would be extremely difficult for a simple hand or finger blob tracker. Our experiments highlighted the strengths of our system:

- **Robustness to cluttered backgrounds.** Our system finds the optimal model registration between image features and states in the DHSSM or the unique clutter state q_c. By incorporating q_c into the registration formulation, our detection method becomes powerful and can handle heavy clutter. Success in scenes with clutter was not demonstrated for an existing real-time hand tracker [10] that was based on matching between boundary templates and object silhouettes.
- **Robustness to illumination and background changes.** Our online learning step updates the classifier so that it can best discriminate the object boundary from clutter. The need for online learning was shown most succinctly in our experiments where the appearance of the object color changed due to lighting changes. Such changes are captured by the DHSSM update.
- **Handling occlusions.** The DP registration process is decomposed into two stages. The first DP stage aims at detecting boundary segments that correspond to states of the DHSSM. Missing segments can be detected during this stage, so that they are not considered in the second stage during which boundary segments are connected to create the boundary of the whole hand.
- **Computational efficiency.** We exploit the spatial coherence between features obtained from temporal information so that we can significantly reduce the size of the DP table. The average processing time per frame for 160×120 images is 1–2 s and the SVM training step takes typically 3–5 s. This compares favorably to other tracking approaches that use graphical models and for which running times of several minutes per frame were reported [6].

Our future work will extend the DHSSM framework to enable us to insert or delete model states through time. Unlike the current version, where every possible structure change is described by the DHSSM, a more powerful DHSSM should be able to learn unpredicted object deformations online. This may allow us to apply the model to complicated deformable objects, such as cells in microscopic images.

Acknowledgment. This research was supported in part by NSF grants IIS-0705749 and IIS-0713229.

References

1. Bar-Shalom, Y., Fortmann, T.: Tracking and Data Association. Academic Press, London (1988)
2. Athitsos, V., Wang, J., Sclaroff, S., Betke, M.: Detecting instances of shape classes that exhibit variable structure. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 121–134. Springer, Heidelberg (2006)

3. Wang, J., Athitsos, V., Sclaroff, S., Betke, M.: Detecting objects of variable shape structure with hidden state shape models. IEEE T PAMI 30(3), 477–492 (2008)
4. Han, B., Zhu, Y., Comaniciu, D., Davis, L.: Kernel-based Bayesian filtering for object tracking. In: CVPR, vol. 1, pp. 227–234 (2005)
5. MacCormick, J., Isard, M.: Partitioned sampling, articulated objects, and interface-quality hand tracking. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 3–19. Springer, Heidelberg (2000)
6. Sudderth, E., Mandel, M., Freeman, W., Willsky, A.: Distributed occlusion reasoning for tracking with nonparametric belief propagation. In: NIPS (2004)
7. Sigal, L., Bhatia, S., Roth, S., Black, M.J., Isard, M.: Tracking loose-limbed people. In: CVPR, pp. 421–428 (2004)
8. Athitsos, V., Sclaroff, S.: Estimating 3d hand pose from a cluttered image. In: CVPR, pp. 432–440 (2003)
9. Shakhnarovich, G., Viola, P., Darrell, T.: Fast pose estimation with parameter-sensitive hashing. In: ICCV, pp. 750–758 (2003)
10. Stenger, B., Thayananthan, A., Torr, P., Cipolla, R.: Hand pose estimation using hierarchical detection. In: Proceeding of International Workshop on Human-Computer Interaction. LNCS (2004)
11. Forsyth, D., Arikan, O., Ikemoto, L., O'Brien, J., Ramanan, D.: Computational Studies of Human Motion: Part 1, Tracking and Motion Synthesis (2006)
12. Ramanan, D., Forsyth, D.A., Zisserman, A.: Strike a pose: Tracking people by finding stylized poses. In: CVPR, pp. 20–25 (2005)
13. Felzenszwalb, P.F., Schwartz, J.D.: Hierarchical matching of deformable shapes. In: CVPR, pp. 1–8 (2007)
14. Opelt, A., Pinz, A., Zisserman, A.: A boundary-fragment-model for object detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 575–588. Springer, Heidelberg (2006)
15. Shotton, J., Blake, A., Cipolla, R.: Contour-based learning for object detection. In: ICCV, pp. 503–510 (2005)
16. Heap, T., Hogg, D.: Wormholes in shape space: Tracking through discontinuous changes in shape. In: ICCV, pp. 344–349 (1998)
17. Rabiner, L.R.: A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceedings of the IEEE 77(2), 257–286 (1989)
18. Collins, R., Liu, Y.: On-line selection of discriminative tracking features. In: ICCV, pp. 346–354 (2003)
19. Schmidt, F.R., Farin, D., Cremers, D.: Fast matching of planar shapes in sub-cubic runtime. In: ICCV, pp. 1–6 (October 2007)

Interactive Tracking of 2D Generic Objects with Spacetime Optimization

Xiaolin K. Wei and Jinxiang Chai

Texas A&M University
xwei@cs.tamu.edu, jchai@cs.tamu.edu

Abstract. We present a continuous optimization framework for interactive tracking of 2D generic objects in a single video stream. The user begins with specifying the locations of a target object in a small set of keyframes; the system then automatically tracks locations of the objects by combining user constraints with visual measurements across the entire sequence. We formulate the problem in a spacetime optimization framework that optimizes over the whole sequence simultaneously. The resulting solution is consistent with visual measurements across the entire sequence while satisfying user constraints. We also introduce prior terms to reduce tracking ambiguity. We demonstrate the power of our algorithm on tracking objects with significant occlusions, scale and orientation changes, illumination changes, sudden movement of objects, and also simultaneous tracking of multiple objects. We compare the performance of our algorithm with alternative methods.

1 Introduction

Our objective in this paper is to track generic moving objects, including 2D locations, 2D scales and orientation, from a monocular video sequence. Building an interactive, accurate object tracking system is challenging because appearances of objects might change overtime due to occlusions, object deformations and illumination changes. Background clutter and sudden movements of the camera or object might further deteriorate the performance of a tracking system.

One solution to this problem is sequential object tracking, which initializes a tracker in one frame and then performs tracking forward recursively in time [1,2]. The approach is computationally efficient for online applications but might not be appropriate for many offline applications such as video-based motion capture, object-based video annotation, compression, and editing, where tracking accuracy is more preferred than real time performance. We believe an ideal system for offline object tracking should take full use of image measurements. When tracking goes wrong, user interaction is also needed to correct errors. Therefore, another desirable feature for offline tracking is to allow the user to specify constraints throughout the video to remove tracking ambiguities. The "feed forward" approach [1,2], however, does not consider frames and user constraints in the future.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 657–670, 2008.

Another solution is to formulate offline tracking in a spacetime optimization framework [3] and consider the entire motion simultaneously [4,5,6]. The approach is appealing because the system can use image measurements and user constraints in both past and future to estimate the current state. Previous work in this direction often use discrete optimization to compute the optimal solution. However, as the number of frames or dimensions of the object state increase, discrete optimization might be too expensive to be conducted. Therefore previous approaches either use expensive preprocessing [4] or an object detector trained offline [5,6] to reduce the search space.

In this paper, we present a continuous trajectory optimization framework for offline tracking of 2D generic objects. Our optimization is very efficient; the speed of the spacetime tracker is comparable to that of sequential object tracking such as meanshift [1]. The system also does not require any preprocessing and offline learning steps. Other benefits of the framework are automatic occlusion handling with robust statistics, explicit modeling of illumination changes with weighted template models, ambiguity reduction with motion priors, and simultaneous tracking of multiple object with group priors.

We demonstrate the power and flexibility of this approach by interactive tracking of cars or sports players in various difficult video sequences. We show the system can accurately track a 2D generic object or or a group of objects with a minimal amount of user interaction. We also show its robustness to occlusions, background clutter, lighting and scale changes. Finally, we compare alternative techniques for 2D object tracking, including mean shift [2] and particle filter [1]. The experiment shows that the spacetime tracker can produce more accurate results with comparable processing time as sequential trackers.

2 Background

In this section, we discuss related work in tracking 2D generic objects from video. Previous work in 2D object tracking can be classified into three main categories: recursive tracking [2,7,8,9,1,10,11,12,13,14], batch-based optimizations [15,4], and tracking-by-detection [16,5,6].

Recursive object tracking methods initialize one frame and then recursively track the evolution of the state forward in time. For example, kernel-based methods sequentially track the state of nonrigid objects by minimizing viewpoint-insensitive histograms between consecutive frames [2,7,13,14]. Kalman filter extensions achieve more robust tracking of maneuvering objects by introducing statistical models of object or camera motion [8,9]. Tracking through occlusion and clutter is achieved by reasoning over a state-space of multiple hypotheses via sequential monte carlo sampling methods [1,10,12].

In contrast, batch-based optimization approach formulates the tracking problem as a trajectory optimization and computes the entire motion sequence simultaneously. In particular, Sun and his colleagues explored discrete optimization to track the object state (2D positions and 1D uniform scale) throughout the video [4]. To reduce the search space for discrete optimization, they first run

the mean shift algorithm to identify 2D candidate regions of the object in each frame, and then applied spectral clustering to extract a number of 3D trajectory segments of the object. Our approach also uses batch-based optimization for 2D object tracking. Unlike previous approaches, we formulate the problem in a continuous optimization framework. The continuous optimization allows us to simultaneously track a group of objects, a capacity that has not been demonstrated in discrete optimization approaches. More importantly, our continuous optimization does not require any preprocessing steps.

Another approach for 2D object tracking is to apply object detection algorithms for tracking. For example, researchers have adapted SVM recognition algorithms for efficient visual tracking [16]. Recently, detection and optimization have also been combined to obtain some of the advantages of each approach [5,6]. However, detection approaches require an offline training step for particular objects, and might not be appropriate for tracking an arbitrary 2D objects.

Our approach is motivated by keyframe guided rotoscoping [15] which uses continuous optimization to track contours of an object with user-specified contours in two or more keyframes. Rotoscoping makes full use of the information in the keyframes to improve the performance of contour tracking. Our work extends this approach significantly by applying it to 2D generic objects.

3 Overview

We formulate object tracking as a spacetime optimization problem, which optimizes over entire sequence simultaneously under user-specified constraints. The system contains three major components:

User interactions. The user begins by clicking the location of target objects in the first and the last frame. The keyframes are then used to interpolate inbetween motions based on image measurements of the entire sequence and predefined priors.

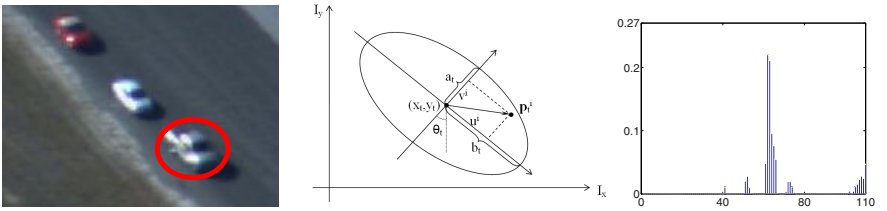
Spacetime optimization. The system computes the "best" motion by optimizing over the whole image sequence at once. The objective function includes keyframe constraints, a weighted template model, an image measurement term and motion prior terms.

Refinement. Because of the difficulty of the tracking problem, there will often be unsatisfactory aspects of the tracking results. Once the optimization has completed, the user may refine tracking results in any frame and then rerun the optimization with these new constraints.

We define the feature model of target objects in the next section and then discuss the spacetime tracking in detail in section 5.

4 Target Representation

In this section, we first discuss how to define a feature space to characterize target objects (section 4.1). We then describe how to represent the template of



(a) (b) (c)

Fig. 1. Target representations: (a) A target object is approximated as an elliptic region; (b) The elliptic region is parameterized by 5-dimensional vector $\mathbf{z}_t = (x_t, y_t, a_t, b_t, \theta_t)^T$; (c) The appearance of the object can thus be represented using histogram distributions of color information in the HSV color space

target objects with weighted template models (section 4.2). We also show how to evaluate the distance between the target template and target candidate (section 4.3).

4.1 Feature Space

In our experiments, a target object is approximated by an elliptic region (see Figure 1(a)). The state of the target object at frame t can thus be described by parameters of the ellipse (see Figure 1(b)):

$$\mathbf{z}_t = (x_t, y_t, a_t, b_t, \theta_t)^T \tag{1}$$

where the parameters x_t and y_t represent the coordinates of the center of the ellipse and the parameters a_t and b_t specify the lengths of the long and short axes respectively. The parameter θ_t is the orientation of the ellipse.

We choose the Hue-Saturation-Value (HSV) color space to model the appearance of target objects. We further define the feature model of a target object as a histogram distribution of all pixels within the elliptic region (see Figure 1(c)). The feature model of a target object at frame t is represented by

$$\mathbf{h}(\mathbf{z}_t) = (h_1(\mathbf{z}_t), \ldots, h_M(\mathbf{z}_t))^T, \quad \sum_{m=1}^{M} h_m(\mathbf{z}_t) = 1 \tag{2}$$

where the parameter M is the total number of bins used in the HSV color space and the function $h_m(\mathbf{z}_t)$ is the density of the m-th bin. Let n_t be the number of pixels located inside the target region at frame t and $\mathbf{p}_t^i, i = 1, ..., n_t$ be the image coordinates of the i-th pixel. Mathematically, we can define the function $h_m(\mathbf{z}_t)$ as follows:

$$h_m(\mathbf{z}_t) = \sum_{i=1}^{n_t} \delta(f(I(\mathbf{p}_t^i)) - m) \tag{3}$$

where the function $\delta(\cdot)$ represents the Kronecker delta function. The function $I(\mathbf{p}_t^i)$ represents the color of the i-th pixel at the location $\mathbf{p}_t^i$. The function f maps $I(\mathbf{p}_t^i)$ to the index of its bin in the quantized feature space.

Because color information is only reliable when both the saturation and the value are not too small, we populate an HS histogram with $N_h N_s$ bins using

only the pixels with saturation and value larger than 0.1 and 0.2 respectively. The remaining "color-free" pixels can however retain a crucial information when tracked regions are mainly black and white. We thus use N_v bins to quantize the V space separately. The resulting complete histogram is composed of $M = N_h N_s + N_v$ bins. In our experiments, N_h, N_s and N_v are set to 10 experimentally. Figure 1(c) shows a histogram distribution computed from color information of a target object shown in Figures 1(a).

We regularize the histogram distribution $h_m(\mathbf{z}_t)$ by masking the objects with an isotropic kernel in the spatial domain. When the kernel weights, carrying continuous information, are used in defining the feature space representation, the regularized histogram distribution of target regions becomes a smooth and continuous function of target states, $\mathbf{z}_t$.

An isotropic kernel, with a convex and monotonic decreasing kernel function $k(r)$, is used to assign smaller weights to pixels farther from the center. In our experiments, we choose the Epanechnikov profile as our kernel function [17]:

$$k(r) = \begin{cases} 1 - r & 0 \le r \le 1 \\ 0 & r > 1 \end{cases} \tag{4}$$

where $r \ge 0$. This kernel function makes the regularized histogram distribution differentiable everywhere inside the elliptical region. Its gradients can, therefore, be evaluated analytically.

The regularized histogram distribution of the feature in the target region at frame t is computed as:

$$h_m(\mathbf{z}_t) = \frac{\sum_{i=1}^{n_t} k\left((\frac{u^i}{a})^2 + (\frac{v^i}{b})^2\right) \delta\big(f(I(\mathbf{p}_t^i)) - m\big)}{\sum_{i=1}^{n_t} k\left((\frac{u^i}{a})^2 + (\frac{v^i}{b})^2\right)} \tag{5}$$

The parameters u^i and v^i are the local coordinates of the i-th pixel (see Figure1(b)), which can be computed as follows:

$$\begin{pmatrix} u^i \\ v^i \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} (\mathbf{p}_t^i - \mathbf{c}_t) \tag{6}$$

where the vector $\mathbf{c}_t$ is a 2D vector (x_t, y_t). Therefore, the feature model of a target region $\mathbf{h}(\mathbf{z}_t)$ is a vector-valued function of the object state $\mathbf{z}_t = (x_t, y_t, a_t, b_t, \theta_t)^T$.

4.2 Weighted Template Model

Our interactive tracking system starts with keyframe constraints defined at the first and last frame. Let $\mathbf{z}_1$ and $\mathbf{z}_T$ represent the state of the first and last frame respectively. We start by computing the feature models of the first and last frame $\mathbf{h}(\mathbf{z}_1)$ and $\mathbf{h}(\mathbf{z}_T)$.

We assume that the template model for any in-between frames can be modeled as a linear interpolation of the feature models at keyframes:

$$H_m(\beta_t) = \beta_t h_m(\mathbf{z}_1) + (1 - \beta_t) h_m(\mathbf{z}_T) \qquad 0 \le \beta_t \le 1,\ 1 < t < T \tag{7}$$

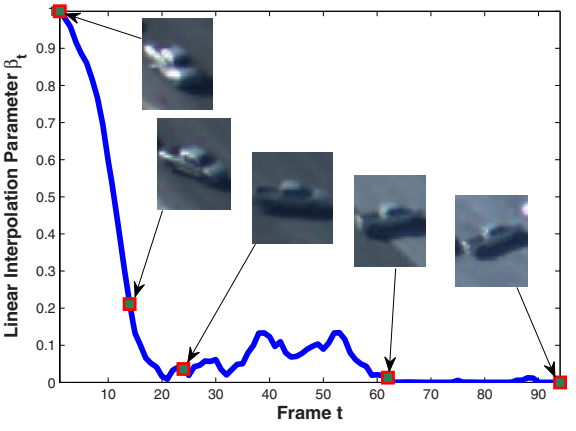


Fig. 2. The optimized β_t over time for the "Car Team" sequence: The red points are the optimized β values for five frames shown on the top respectively

where the interpolation weight β_t is a scalar value between 0 and 1. The interpolated feature model at frame t, $\mathbf{H}(\beta_t) = (H_1(\beta_t), \ldots, H_M(\beta_t))^T$, is therefore a function of β_t. One attractive feature of the weighted template models is to model possible appearance changes between two keyframes with the parameter β_t. For example, when the weight β_t is close to one, the appearance of the template object becomes similar to that of the first keyframe. When the weight β_t approaches to zero, the template object looks more like the target object in the last keyframe.

Figure 2 shows the evolution of the weight β_t optimized from one testing video sequence. The images on the top show five sample frames of the target vehicle. The appearances of the target object is consistent with the reconstructed β_t values (red points).

4.3 Similarity Function

The similarity function is used to evaluate the distance between a template object and candidate objects in the feature space.

We define their distance, d, as

$$d(\mathbf{z}_t, \beta_t) = \sqrt{1 - \sum_{m=1}^{M} \sqrt{h_m(\mathbf{z}_t)\, H_m(\beta_t)}} \tag{8}$$

where the function $h_m(\mathbf{z}_t)$ is the feature model of an candidate object and the function $H_m(\beta_t)$ is the weighted template model at frame t. Both functions are regularized histogram distributions. A differential kernel function yields a differentiable similarity function. We, therefore, can evaluate the gradient of the similarity function in terms of β_t and $\mathbf{z}_t$ analytically.

5 Spacetime Object Tracking

We formulate object tracking as a continuous trajectory optimization problem and estimate the states of objects across the entire sequence simultaneously. This section discusses how to derive the objective function and how to optimize it efficiently.

5.1 Objective Function

The objective function consists of two types of energy terms: *image* term and *prior* term. The image term prefers a solution which minimizes appearance difference between the template object and candidate object. The prior term penalizes quickly moving objects and sudden changes of scales and appearances. The prior term is pivotal for removing the tracking ambiguity due to occlusions or clutter.

Image Term. The data term, E_I, measures similarity between the template model $\mathbf{H}(\beta_t)$ and candidate object $\mathbf{h}(\mathbf{z}_t)$ ranging from frame 2 to frame $T-1$.

To deal with occlusions, we apply robust statistics to measure the data term. Robust estimation addresses the problem for finding the values for the parameters from the measurement data with outliers, which correspond to heavily occluded objects in our experiments. We therefore can define the data term as follows:

$$E_I = \sum_{t=2}^{T-1} \rho(d(\mathbf{z}_t, \beta_t)) \tag{9}$$

To increase robustness we will consider estimators for which the influence of outliers tends to zero. We choose the Lorentzian estimator but the treatment here could equally be applied to a wide variety of the other estimator. A discussion of various estimators can be found in [18,19]. More specifically, the Lorentzian function is defined as follows:

$$\rho(r) = \log(1 + 1/2(r/\sigma)^2) \tag{10}$$

where the scalar σ is a parameter for the robust estimator. In our experiments, σ is set to 0.2475.

Prior Terms. Due to occlusions, background clutter, illumination or scale changes, image term might not be sufficient for tracking objects correctly. Prior term is very important for reducing the tracking ambiguity. There are two types of prior terms used in our paper:

- The state smoothness term, E_P, minimizes state changes over time:

$$E_P = \sum_{t=2}^{T} \|\mathbf{z}_t - \mathbf{z}_{t-1}\|^2 \tag{11}$$

  where $\mathbf{z}_t$ is the state of the target object at frame t, which includes 2D location, 2D scales, and orientation.

– The illumination smoothness term, E_L, measures the temporal smoothness of the interpolation weight, β_t, over time.

$$E_L = \sum_{t=2}^{T} (\beta_t - \beta_{t-1})^2 \tag{12}$$

Tracking objects only by prior terms is similar to performing smooth interpolation of keyframes.

5.2 Optimization Method

After combining the image term and prior terms, our interactive object tracking problem becomes the following constrained nonlinear optimization problem:

$$\begin{array}{c} arg\, min_{\mathbf{Z},\beta} \;\; E_I(\mathbf{Z},\beta) + \lambda_P E_P(\mathbf{Z}) + \lambda_L E_L(\beta) \\ s.t. \;\; 0 \le \beta_t \le 1 \;\; t = 1, ..., T \end{array} \tag{13}$$

where $\mathbf{Z}$ and β are the concatenation of the system states $\mathbf{z}_t$ and the concatenation of the interpolation weights β_t from frame 2 to frame $T-1$. The weights λ_P and λ_R are set to 0.0015 and 0.5 respectively.

The number of variables in our optimization can be quite large — about five times the number of total frames. The Jacobian matrix for our objective function, however, is very sparse. We therefore optimize the objective function with large-scale trust-region reflective Newton methods.

A linear interpolation of the user-specified keyframes is used for optimization initialization. We found that the optimization procedure runs very efficiently and always converges quickly (usually less than 20 iterations). In our experiments, we found that the spacetime tracker with a Matlab implementation can track objects interactively (from 12 fps to 25 fps).

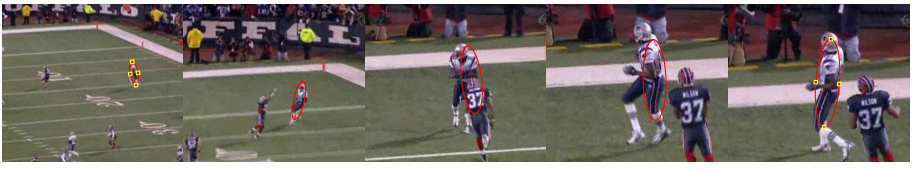
The gradient of the energy function is analytically evaluated at each iteration (See Appendix). The two prior terms, Equation 11 and Equation 12, are quadratic functions, whose derivatives can be easily derived. The partial derivatives of the image term can also be evaluated analytically because the similarity function is analytically differentiable (See Appendix).

Due to the complexity of a real world, it is almost impossible to build a fully automatic system that can robustly track any interesting objects from video. When a tracker fails, user interaction must be used to correct tracking errors. Our system provides an efficient way to combine user interactions with an automatic vision process. The user can refine the tracking result at any frame and restart the optimization. When more than two keyframes are defined, the spacetime solver computes the solution for each subsequence separately.

Like any gradient-based optimization methods, the system might fall in local minima. Though, we rarely have this problem in our experiments. One strength of the system is to involve the user into the loop. If the optimization falls in a local minima, the user can briefly review the result and then specify more keyframes to rerun the optimization.

Table 1. The statistics for video sequences tested in our paper

Sequence	Length	Occlusions	Scale changes	Lighting changes	Running Time
Touchdown	100	large	large	none	11.9s
IR	240	partial	small	large	14.9s
Rushing	94	large	small	none	9.4s
Truck in Forest	130	large	medium	large	22.4s
Car Teams (one car)	94	partial	small	medium	6.3s



#1 #45 #75 #90 #100

Fig. 3. The "Touchdown" sequence shows our algorithm's robustness to scale changes

5.3 Multi-object Tracking

Our framework can also be used to track a group of objects simultaneously. When multiple objects move as a group such as a group of racing cars or football players, their movements are not independent. One advantage of simultaneous tracking of multiple objects is to utilize motion prior between multiple objects to reduce tracking uncertainty.

Similarly, we can formulate the multi-object tracking problem as the following optimization problem:

$$argmin_{\{\mathbf{Z}^n,\beta^n\}} \sum_n (E_I^n(\mathbf{Z}^n,\beta^n)+\lambda_P E_P(\mathbf{Z}^n) + \lambda_L E_L(\beta^n)) + \lambda_G E_G(\mathbf{Z}^1,...,\mathbf{Z}^N)$$
$$s.t. \quad 0 \leq \beta_t^n \leq 1 \quad t = 1,...,T \tag{14}$$

where N is the total number of objects to be tracked. The parameters $\mathbf{Z}^n$ and β^n are the concatenation of the system states $\mathbf{z}_t$ and the concatenation of the interpolation weights β_t over time for the n-th object. The optimization term E_G models group prior for multi-object movements. In our experiments, we choose to minimize the changes of the relative distances between multiple objects, though more sophisticated group priors might be used.

6 Experimental Results

In this section, we first test the performance of our algorithm on real video sequences. We then compare our algorithm with two alternative object tracking techniques–particle filter [1] and mean shift [2]. A short summary of testing video sequences and their computational time is reported in Table 1. All the computational time reported here is based on Matlab implementations. Our tracking results are best seen in the accompanying video.

6.1 Testing on Real Video

We tested the effectiveness of our system on different kinds of video sequences with occlusions, scale changes, and illumination changes (see Figures 3, 4, 5, and 6). We have also done experiments on simultaneous tracking of multiple objects (see Figure 7).

The "Touchdown" sequence contains a target object with significant scale changes (see Figure 3). Throughout the testing sequence, the scales of the player increase from $(a_1, b_1) = (10.1, 22.5)$ to $(a_{100}, b_{100}) = (45.1, 117.7)$ because of the zooming of a camera.

The "IR" sequence has significant illumination changes. The low-resolution and noisy video makes the tracking problem even more challenging. Figure 4 shows our system can track the video accurately.

The "Rushing" sequence demonstrates the performance of our algorithm on tracking target objects with significant occlusions (see Figure 5). The video has significant occlusions in two different places. From frame 15 to 27, the target player is occluded by an opponent player in a different uniform. The second occlusion lasts longer (from frame 59 to 81); the target player is completely occluded by his teammate with the same uniform during a certain period of time.

The "Truck in Forest" sequence demonstrates the performance of our algorithm on tracking target objects with significant occlusions, illumination changes (shadows) and orientation changes (see Figure 6). In addition, the robust estimator for the data term allows the system to automatically detect when the object is heavily occluded (see blue ellipses).

Fig. 4. The "IR" sequence includes significant illumination changes and is successfully tracked by our algorithm

Fig. 5. The "Rushing" sequence includes significant occlusions and is successfully tracked by our algorithm

Fig. 6. The "Truck in Forest" sequence shows the performance of our algorithm on tracking an object with significant occlusions. The system can also detect when the target is occluded (in blue ellipses).

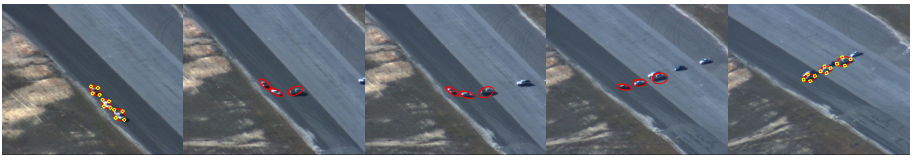


Fig. 7. Simultaneous tracking of three cars with group priors

The "Car Team" sequence demonstrates the power of our algorithm on tracking multiple objects simultaneously (see Figure 7). It is a difficult sequence to track. The cars have a sudden turn in the middle of the sequence and target objects also look very similar. With the group prior described in section 5.3, we successfully track the movements of three cars in the same team.

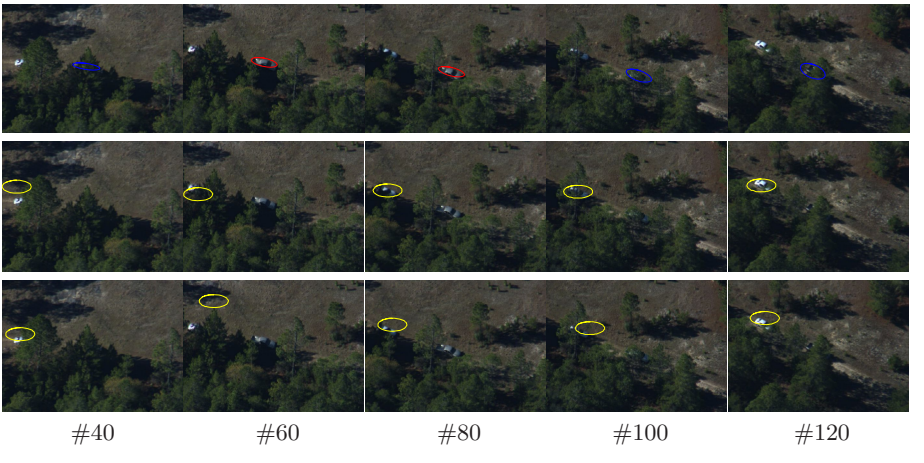


Fig. 8. Comparisons on the "Truck in Forest" sequence. Our spacetime tracker (first row) successfully tracked the whole sequence. The particle filter (second row) and the mean shift method (third row) failed to track the object across the entire sequence.

Table 2. The number of keyframes needed and the running time (in brackets) for different methods. The running time is based on Matlab implementations.

Sequence	Mean shift	Particle filtering	Our method
Touchdown	4 (214s)	5 (74s)	2 (12s)
IR	6 (31s)	5 (19s)	2 (15s)
Rushing	6 (33s)	6 (19s)	2 (9s)
Truck in Forest	9 (180s)	8 (52s)	2 (22s)
Car Teams (one car)	5 (49s)	4 (27s)	2 (6s)

6.2 Comparisons with Other Methods

This section compares the performance of our spacetime tracking algorithm with particle filter [1] and mean shift [2]. Figure 8 shows comparison results for the "Truck in Forest" sequence. Table 2 reports, for different methods, the number of keyframes and the computational time needed in order to track the targets successfully. Our method provides better performance than the two alternative algorithms.

7 Discussion

We have presented a spacetime optimization approach for 2D generic object tracking. Unlike previous offline tracking algorithms [4,5,6], our system uses continuous optimization for 2D object tracking and does not require any preprocessing or off-line learning learning steps. The system can achieve high quality results with minimal user input. Our experiments demonstrate the power of our algorithm on tracking objects with occlusions, scale changes, illumination changes, and sudden movement of objects. We also show the performance algorithm on simultaneous tracking of multiple objects which might be too expensive for previous offline tracking algorithms.

As compared to recursive tracking methods, our method is more appropriate for interactive applications such as object-based video annotation, compression, editing, and video based motion capture. For those applications, high quality results are far more important than real-time performances. In the mean time, all video frames are available in advance. Combining user interaction with image measurements across the entire video sequence provides more accurate results than automatic tracking methods.

We believe the basic idea of our spacetime tracking could also be used for tracking other types of objects. One of immediate directions for future work is, therefore, to extend our spacetime optimization for interactive feature tracking and articulated object tracking.

References

1. Isard, M., Blake, A.: Condensation conditional density propagation for visual tracking. International Journal on Computer Vision 29(1), 5–28 (1998)
2. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Trans. Pattern Analysis and Machine Intelligence 25(3), 564–577 (2003)

3. Witkin, A., Kass, M.: Spacetime constraints. In: Proceedings of ACM SIGGRAPH 1998, pp. 159–168 (1988)
4. Sun, J., Zhang, W., Tang, X., Shum, H.Y.: Bi-directional tracking using trajectory segment analysis. In: Proceedings of ICCV, vol. 1, pp. 717–724 (2005)
5. Buchanan, A., Fitzgibbon, A.: Interactive feature tracking using k-d trees and dynamic programming. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 626–633 (2006)
6. Wei, Y., Sun, J., Tang, X., Shum, H.Y.: Interactive offline tracking for color object. In: Proceedings of ICCV (2007)
7. Elgammal, A., Duraiswami, R., Davis, L.: Probabilistic tracking in joint feature-spatial spaces. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 781–788 (2003)
8. Blackman, S., Popoli, R.: Design and analysis of modern tracking systems. Artech House Publishers (1999)
9. Koller, D., Daniilidis, K., Nage, H.: Model-based object tracking in monocular image sequences of road traffic scenes. International Journal on Computer Vision 10(3), 257–281 (1993)
10. Rasmussen, C., Hager, G.: Joint probabilistic techniques for tracking multi-part objects. IEEE Trans. Pattern Analysis and Machine Intelligence 23, 560(n)-576 (1993)
11. Jepson, A., Fleet, D., El-Maraghi, T.: Robust online appearance models for visual tracking. IEEE Trans. Pattern Analysis and Machine Intelligence 25(10), 1296–1311 (2003)
12. Wu, Y., Huang, T.S.: Robust visual tracking by integrating multiple cues based on co-inference learning. International Journal on Computer Vision 58(1), 55–71 (2004)
13. Guskov, I.: Kernel-based template alignment. In: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 610–617 (2006)
14. Megret, R., Mikram, M., Berthoumieu, Y.: Inverse composition for multi-kernel tracking. In: Computer Vision, Graphics and Image Processing. LNCS, pp. 480–491 (2007)
15. Agarwala, A., Hertzmann, A., Salesin, D.H., Seitz, S.M.: Keyframe-based tracking for rotoscoping and animation. ACM Transactions on Graphics 24(3), 584–591 (2005)
16. Avidan, S.: Support vector tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(8), 1064–1072 (2004)
17. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Analysis and Machine Intelligence 24(5), 603–619 (2002)
18. Huber, P.: Robust statistics. Wiley, Chichester (1981)
19. Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J., Stahel, W.A.: Robust statistics: The approach based on influence functions. Wiley, Chichester (1986)

Appendix

This appendix shows how to derive the Jacobian matrix of Equation 8. The Jacobian matrix can be analytically evaluated based on the following partial derivatives.

$$\frac{\partial d(\mathbf{z}_t,\beta_t)}{\partial x_t} = -\frac{1}{a_t^2}F(\mathbf{z}_t,\beta_t)\sum_{m=1..M}^{h_m(\mathbf{z}_t)\neq 0}\overline{\frac{H_m(\beta_t)}{h_m(\mathbf{z}_t)}}\sum_{i=1}^{n_t}u_t^i\quad \delta_m(\mathbf{p}_t^i)-h_m(\mathbf{z}_t) \tag{15}$$

$$\frac{\partial d(\mathbf{z}_t,\beta_t)}{\partial y_t} = -\frac{1}{b_t^2}F(\mathbf{z}_t,\beta_t)\sum_{m=1..M}^{h_m(\mathbf{z}_t)\neq 0}\overline{\frac{H_m(\beta_t)}{h_m(\mathbf{z}_t)}}\sum_{i=1}^{n_t}v_t^i\quad \delta_m(\mathbf{p}_t^i)-h_m(\mathbf{z}_t) \tag{16}$$

$$\frac{\partial d(\mathbf{z}_t,\beta_t)}{\partial a_t} = -\frac{1}{a_t^3}F(\mathbf{z}_t,\beta_t)\sum_{m=1..M}^{h_m(\mathbf{z}_t)\neq 0}\overline{\frac{H_m(\beta_t)}{h_m(\mathbf{z}_t)}}\sum_{i=1}^{n_t}(u_t^i)^2\quad \delta_m(\mathbf{p}_t^i)-h_m(\mathbf{z}_t) \tag{17}$$

$$\frac{\partial d(\mathbf{z}_t,\beta_t)}{\partial b_t} = -\frac{1}{b_t^3}F(\mathbf{z}_t,\beta_t)\sum_{m=1..M}^{h_m(\mathbf{z}_t)\neq 0}\overline{\frac{H_m(\beta_t)}{h_m(\mathbf{z}_t)}}\sum_{i=1}^{n_t}(v_t^i)^2\quad \delta_m(\mathbf{p}_t^i)-h_m(\mathbf{z}_t) \tag{18}$$

$$\frac{\partial d(\mathbf{z}_t,\beta_t)}{\partial \theta_t} = (\frac{1}{a_t^2}-\frac{1}{b_t^2})F(\mathbf{z}_t,\beta_t)\sum_{m=1..M}^{h_m(\mathbf{z}_t)\neq 0}\overline{\frac{H_m(\beta_t)}{h_m(\mathbf{z}_t)}}\sum_{i=1}^{n_t}u_t^iv_t^i\quad \delta_m(\mathbf{p}_t^i)-h_m(\mathbf{z}_t) \tag{19}$$

$$\frac{\partial d(\mathbf{z}_t,\beta_t)}{\partial \beta_t} = \frac{1}{4d(\mathbf{z}_t,\beta_t)}\sum_{m=1..M}^{H_m(\beta_t)\neq 0}\overline{\frac{h_m(\mathbf{z}_t)}{H_m(\beta_t)}}\,(H_m(\beta_t)-H_m(\beta_1)) \tag{20}$$

where,

$$F(\mathbf{z}_t,\beta_t)=\frac{1}{2d(\mathbf{z}_t,\beta_t)\sum_{j=1}^{n_t}k\left((u_t^i/a_t)^2+(v_t^i/b_t)^2\right)}\quad\text{and}\quad \delta_m(\mathbf{p}_t^i)=\delta(f(I(\mathbf{p}_t^i))-m)$$

Note that, in Equation 15, 16, 17, 18 and 19, we should drop off the components when $h_m(\mathbf{z}_t)=0$ in the summations over m. Because when $h_m(\mathbf{z}_t)=0$, $h_m(\mathbf{z}_t)$ will not change with respect to $\mathbf{z}_t$ (see Equation 5). Thus partial derivatives of $\sqrt{h_m(\mathbf{z}_t)H_m(\beta_t)}$ (in Equation 8) with respect to $\mathbf{z}_t$ are zeros, so the m's with $h_m(\mathbf{z}_t)=0$ should be dropped off. Similarly, the components with $H_m(\beta_t)=0$ in Equation 20 also need to be dropped off.

The distance function (Equation 8) is differentiable everywhere inside the elliptical region. Therefore the whole objective function is also differentiable inside the elliptical region.

A Segmentation Based Variational Model for Accurate Optical Flow Estimation

Li Xu, Jianing Chen, and Jiaya Jia

Department of Computer Science and Engineering
The Chinese University of Hong Kong
{xuli,jnchen,leojia}@cse.cuhk.edu.hk

Abstract. Segmentation has gained in popularity in stereo matching. However, it is not trivial to incorporate it in optical flow estimation due to the possible non-rigid motion problem. In this paper, we describe a new optical flow scheme containing three phases. First, we partition the input images and integrate the segmentation information into a variational model where each of the segments is constrained by an affine motion. Then the errors brought in by segmentation are measured and stored in a confidence map. The final flow estimation is achieved through a global optimization phase that minimizes an energy function incorporating the confidence map. Extensive experiments show that the proposed method not only produces quantitatively accurate optical flow estimates but also preserves sharp motion boundaries, which makes the optical flow result usable in a number of computer vision applications, such as image/video segmentation and editing.

1 Introduction

Accurate motion estimation is required for solving many computer vision problems, including moving object segmentation and video understanding. However, high-quality motion estimates are usually difficult to be obtained, especially for occluded pixels, discontinuous motion boundaries, and textureless regions.

In stereo matching [1,2,3], color-segmentation-based approaches have demonstrated their strong capability in handling textureless and occluded regions. These methods generally assume a specific (e.g. planar) model for each segment. Regularization is then applied to the model parameters. However, similarly employing the segmentation in modern optical flow frameworks is not easy, owing to the insufficiency of using the segmented regions to constrain the non-rigid motion in consecutive frames. Small-size segments were used in [4] to alleviate this problem. But it usually suffers from the following limitations. For one thing, small segments cannot faithfully represent the structure of natural scenes and thus weaken the regularization power brought forth by segmentation. For another, small patch size might result in poor estimation of the motion parameters due to the local aperture problem.

In this paper, we address two important issues of using segmentation in optical flow - that is, 1) how to know whether the motion model fits the flow in

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 671–684, 2008.

a segment or not, and 2) how to handle the non-rigid motion while faithfully preserving the image structures. To this end, a three-step optical flow framework is proposed. In particular, we first segment the input images based on the color information and the initial motion estimate. The segmentation is then incorporated into a variational model to estimate the motion parameters combining the regularization terms. To reduce the possible segmentation errors caused by inappropriate motion parameterization for non-rigid objects, we compute a confidence map that represents the likelihood whether the parametric flow estimate for a specific pixel is trustworthy or not. This confidence map is taken into a final global optimization step to selectively and locally refine the problematic flow estimates. Our experimental results show that the proposed method is capable of handling both the rigid and non-rigid motions.

2 Related Work

Optical flow is a long studied problem [5,6,7]. Following the framework of Horn and Schunck [6], efforts have been recently put in improving the accuracy and efficiency using a variational model [8,9,10,11,12]. The main issue yet to be addressed is the recovery of high quality motion boundary in the presence of large occlusion.

There exist ways to alleviate the boundary problem. Black and Anandan [7] applied a robust function to handling the possible outliers and the motion discontinuity. This function is also employed in other methods [8,10] and is solved as a modified L-1 norm minimization problem.

Anisotropic diffusion is a method using the gradient information to reduce over-smoothing in the processed region. Tschumperlé *et al.* [13] proposed a matrix-valued scheme for nonlinear diffusion in estimating the matrix-form motion tensor. The adaptive diffusion function returns small values at object boundary, which controls the smoothness over the motion discontinuity. Xiao *et al.* [11] extended the work by substituting the diffusing tensor with an adaptive bilateral filter, and controlled the diffusion process according to the occlusion detection. Although these methods can sharpen the motion boundary, they do not handle well large occlusions, possibly making the recovered motion boundary over-smoothed.

Segmentation- or layer-based approaches assume a parametric motion model (e.g., translational or affine model) for each segment. To handle non-rigid motion, the size of segments has to be small. Zitnick *et al.* [4] generated consistent segments between frames and enforced a translational model within each segment. Piece-wise parametric motion model is also used in [5,7,14] within small patches. This assumption may result in poor estimation of the motion parameters because of the local aperture problem. Black and Jepson [15] relaxed the affine constraint by adding local deformation to the parametric model, resulting in a non-parametric motion. Mémin and Pérez [16] combined the piece-wise parametric motion with local disturbance in a hierarchical setting to mix local flow field with different parameterizations. For these methods, as the model

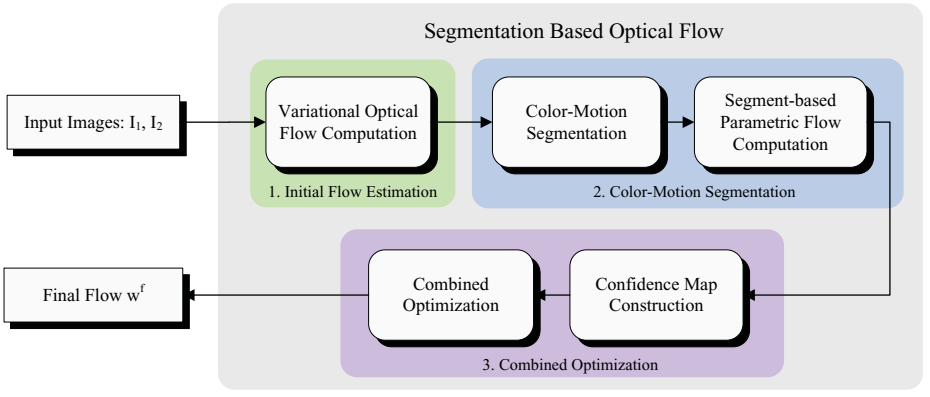


Fig. 1. Overview of our algorithm

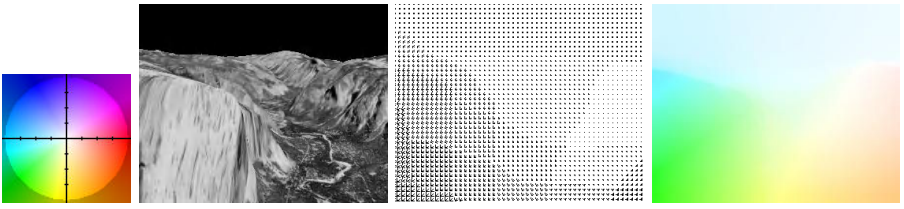
is no longer piece-wise parametric, the regularization power is weakened. The underlying difficulties of the segmentation-based methods are the handling of non-rigid motion and the robust estimation of parameters with the presence of occlusion. These difficulties hinder the widely studied color segmentation from being trivially employed in optical flow estimation.

Another topic related to optical flow is the motion segmentation which aims at extracting moving objects. The segmentation is usually accomplished based on motion [17,18] or the combination of color and motion [19,20,21,22]. Joint estimation of motion and segmentation has recently been studied [17,18,21], where the contour evolving and motion estimation are iteratively performed. The motion segmentation methods cannot be directly applied to flow estimation since extracting a moving object does not need to accurately estimate flow for each and every pixel.

3 Our Approach

Given an image pair (I_1, I_2) that contains objects undergoing small spatially-variant, and possibly non-rigid motion, our objective is to estimate the motion flow vector $\mathbf{w}(\mathbf{x}) = (u(\mathbf{x}), v(\mathbf{x}))^T$ for each pixel $\mathbf{x} = (x, y)^T$ in image I_1. Our approach consists of three main steps. We illustrate the block diagram in Fig. 1. Briefly speaking, we first estimate an initial motion field based on a simple variational model. The initialized flow is then combined with the color information to generate segments. The flow field is refined by segmentation using a robust parameter estimation process. In order to handle the non-rigid motion, a confidence map is constructed measuring the confidence of using the motion parameterization in each segment. The final flow is obtained by relaxing the motion parameterization constraint in a global optimization step using the confidence map.

To visualize the dense flow, in this paper, we adopt the color coding in [23] where the chromaticity is used to distinguish the motion direction and the



(a) (b) (c) (d)

Fig. 2. Flow representation. (a) A reference color wheel. (b) One input image of the *Yosemite* sequence. (c) Traditional flow field representation using arrows. (d) Optical flow field with the color-coded representation.

intensity corresponds to the motion magnitude. Fig. 2 shows the reference color wheel and our *Yosemite* result represented by both flow arrows and colors.

3.1 Initial Flow

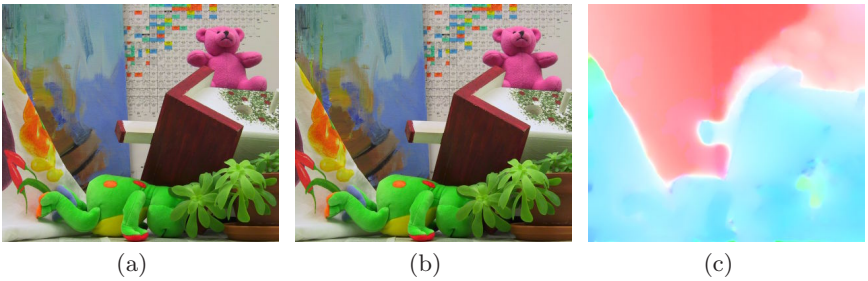
In this stage, we enforce the color constancy and use the variational model similar to that in [8,10] to initialize the flow field. The data term is expressed as

$$E_{Data}(u,v) = \int_{\Omega_1} \sum_{c=1}^{3} \Psi(|I_2(\mathbf{x}+\mathbf{w},c) - I_1(\mathbf{x},c)|^2, \epsilon_D) d\mathbf{x}, \tag{1}$$

where Ω_1 is the domain of image I_1, $\Psi(x,\epsilon)$ is the Total Variation (TV) regularizer [8] defined as $\Psi(x,\epsilon) = \sqrt{x+\epsilon^2}$, and $I(\mathbf{x},c)$ denotes the color of pixel $\mathbf{x}$ in the cth channel of image I. The smoothness term is given by

$$E_{Smooth}(u,v) = \int_{\Omega_1} \Psi(\|\nabla u\|^2 + \|\nabla v\|^2, \epsilon_S) d\mathbf{x}, \tag{2}$$

where ∇ is the first-order derivative operator. In the rest of this paper, we denote $\Psi_D(x) = \Psi(x,\epsilon_D)$ and $\Psi_S(x) = \Psi(x,\epsilon_S)$ for simplicity's sake. The initial flow is estimated by minimizing the combined energy



(a) (b) (c)

Fig. 3. An example of the initial flow. (a) and (b) show the image pair of the "Teddy" example from the Middlebury dataset [23]. (c) The dense flow result obtained in initialization. It is over-smoothed and contains errors around the object boundary.

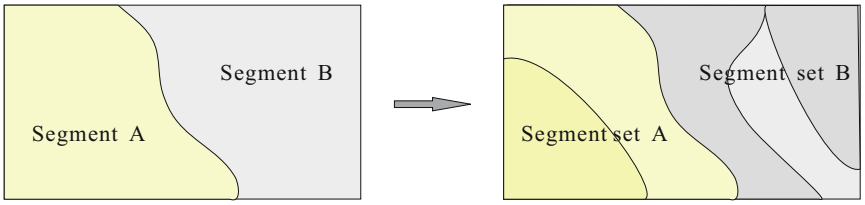


Fig. 4. Two-pass segmentation demonstration

$$E_0(u,v) = E_{Data}(u,v) + \alpha E_{Smooth}(u,v), \tag{3}$$

where α is a weight balancing the two terms. $E_0(u,v)$ is minimized by solving the corresponding Euler-Lagrange equations using the nonlinear multigrid scheme [10]. In this step, we compute the flow bidirectionally, i.e. a flow pair $(\mathbf{w}_1^0, \mathbf{w}_2^0)$, indicating mappings from I_1 to I_2 and from I_2 to I_1 respectively.

This initial flow estimates contain errors mostly in the occluded and textureless regions. One example is shown in Fig. 3 where the flow does not preserve clear edges and the whole map looks over-smoothed. We thus propose incorporating segmentation to improve it.

3.2 Color-Motion Segmentation

In our optical flow framework, the segments are produced counting in both the color and initial flow information, which reduces the possibility of mis-classifying a pixel to a segment only using color. Our method uses a two-pass segmentation scheme which first partitions the input images with regard to color to preserve edge structures, and then further "splits" each segment into more motion-distinctive patches. The two-pass segmentation is performed, in our approach, by the Mean-shift method [24] using the 3D color and 2D flow information respectively. The segmentation parameters all have fixed values in experiments (7 for spatial bandwidth, 6.5/0.02 for the range bandwidth of color/motion, 200 for minimum region size). Fig. 4 illustrates that segments A and B are produced using the color information. They are further split into more motion patches. A merging operation is performed to remove small patches near color discontinuities, as they are typically caused by occlusions and should not be treated as independent segments.

3.3 Parametric Flow Estimating Incorporating Segmentation

Regional information is used in stereo or the motion estimation by assuming a parametric model [7,4] – that is, within each segment, all pixels are in comply with a single translational or affine motion model where the model parameters are estimated either by a regression method [7], or using plane fitting [2,3]. However, these methods have inherent drawbacks when applied to noisy data in dense flow estimation. The regression method is known as possibly suffering from

the local aperture problem. When the size of a segment is not large enough, this approach may lead to an unpredictable parameter estimate. The robust fitting technique depends too much on the initial variable values within each segment. In addition, both of the above methods do not enforce an intra-segment constraint, which is found quite useful in our method to reduce region-wise errors.

We estimate the parametric motions in a regularization framework, similar to that used in [14,16]. Specifically, we define an affine model and denote by $\mathbf{a}_s = (a_{s0}, a_{s1}, a_{s2}, a_{s3}, a_{s4}, a_{s5})^T$ the vector of all affine parameters for segment s. The motion field $\mathbf{w}(\mathbf{a}_s, \mathbf{x}) = (u(\mathbf{a}_s, \mathbf{x}), v(\mathbf{a}_s, \mathbf{x}))^T$ in segment s is given by

$$u(\mathbf{a}_s, \mathbf{x}) = a_{s0}x + a_{s1}y + a_{s2},$$
$$v(\mathbf{a}_s, \mathbf{x}) = a_{s3}x + a_{s4}y + a_{s5}.$$

With the above parametric flow representation in each segment, the energy function w.r.t. all $\mathbf{a}$'s for image I_1 can be written as:

$$E_1(\mathbf{a}) = \int_S \int_s \sum_{c=1}^{3} \Psi_D(|I_2(\mathbf{x} + \mathbf{w}(\mathbf{a}_s, \mathbf{x}), c) - I_1(\mathbf{x}, c)|^2) d\mathbf{x} ds + \alpha \int_{\Omega_1} \Psi_S(\|\nabla u(\mathbf{a}_s, \mathbf{x})\|^2 + \|\nabla v(\mathbf{a}_s, \mathbf{x})\|^2) d\mathbf{x}, \quad (4)$$

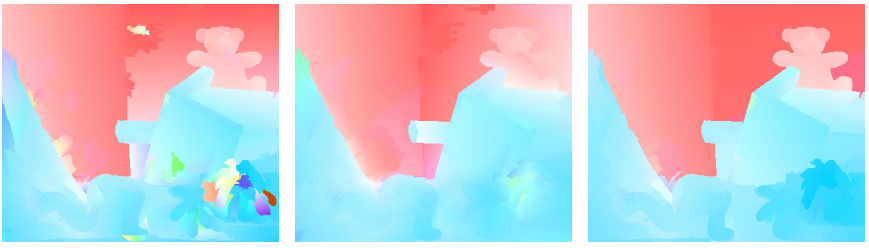
where S is the set of all segments in image I_1. (4) is minimized in a coarse-to-fine manner using a Gaussian pyramid. The affine parameters $\mathbf{a}^{k+1}$ in level $k+1$ are computed by adding increments to the estimated result in level k, i.e., $\mathbf{a}^{k+1} = \mathbf{a}^k + \Delta\mathbf{a}^{k+1}$. In each pyramid level, we use the Taylor expansion to approximate the increments by throwing away high-order terms. This gives us a new increment data term

$$E_{D'}(\Delta\mathbf{a}^{k+1}, \mathbf{x}) = \sum_{c=1}^{3} \Psi_D\big(|(I_x^c)^k \cdot u(\Delta\mathbf{a}_s^{k+1}, \mathbf{x}) + (I_y^c)^k \cdot v(\Delta\mathbf{a}_s^{k+1}, \mathbf{x}) + (I_z^c)^k|^2\big) \quad (5)$$

where $(I_i^c)^k = \partial_i I_2(\mathbf{x} + \mathbf{w}(\mathbf{a}_s^k, \mathbf{x}), c)$, $i = \{x, y\}$, denoting the spatial derivatives. $\mathbf{a}_s^k$ is the affine parameters for segment s estimated in level k. $(I_z^c)^k = I_2(\mathbf{x} + \mathbf{w}(\mathbf{a}_s^k, \mathbf{x}), c) - I_1(\mathbf{x}, c)$. It represents the temporal difference. The smoothness term is written as

$$E_{S'}(\Delta\mathbf{a}^{k+1}, \mathbf{x}) = \Psi_S\big(\|\nabla u(\mathbf{a}_s^k + \Delta\mathbf{a}_s^{k+1}, \mathbf{x})\|^2 + \|\nabla v(\mathbf{a}_s^k + \Delta\mathbf{a}_s^{k+1}, \mathbf{x})\|^2\big), \quad (6)$$

where ∇u and ∇v are approximated by the forward difference. In implementation, the smoothness term is further separated into two parts, i.e., the inter-segment and the intra-segment smoothness w.r.t. the locations of neighboring pixels in computing ∇u and ∇v. The inter-segment smoothness is imposed on the segment boundaries and is used to propagate information among regions. The intra-segment smoothness is enforced within each segment and is uniformly represented as $(a_{s0}^{k+1})^2 + (a_{s1}^{k+1})^2 + (a_{s3}^{k+1})^2 + (a_{s4}^{k+1})^2$. It regularizes the affine parameters and enforces a translational motion model. This is useful for the ubiquitous small-size textureless regions when the data term is not trustworthy.



(a) Robust Regression (b) RANSAC Fitting (c) Our Variational Model

Fig. 5. Comparison of the estimated motion by three methods incorporating the same segmentation information. Our result not only preserves high-quality boundary, but also contains accurate motion estimate within small patches.

The final energy to be minimized in pyramid level $k+1$ combining $E_{S'}$ and $E_{D'}$ is expressed as

$$E(\Delta\mathbf{a}^{k+1}) = \sum_{\mathbf{x}} \left\{ E_{D'}(\Delta\mathbf{a}^{k+1}, \mathbf{x}) + \alpha E_{S'}(\Delta\mathbf{a}^{k+1}, \mathbf{x}) \right\}. \tag{7}$$

Since (7) is continuous, differentiable, and convex with respect to $\Delta\mathbf{a}^{k+1}$, we can use any gradient-based method (our system uses the Quasi-Newton method) to minimize it. In experiments, the flow estimation for each level of the pyramid only needs $5-7$ iterations. The total number of variables (i.e., the affine parameters) to be updated is only $6N_s$ (N_s is the number of segments), which is much smaller than the number of dense flow vectors. So the optimization in this step is efficient and robust.

To demonstrate the effectiveness of the segmentation-combined variational model, we show in Fig. 5 a comparison of the motion results generated by robust regression, robust fitting, and our method based on the same initial flow and segmentation. The robust regression [7] is achieved by removing the regularization term in (4). The plane fitting result is obtained by performing RANSAC [25] for the initial flow to fit the affine parameters. We take 1000 iterations to produce the result. Comparably, these two methods have difficulties to get accurate flow estimates for small segments in the textureless or occluded regions. Our method, on the contrary, takes the advantages of segmentation as well as the intra-segment smoothness, thus can significantly reduce the errors.

3.4 Confidence Map Construction

Segmentation can improve flow estimation in textureless regions. However, it also induces problems for regions undergoing non-rigid motion. In this circumstance, the affine model in each segment could be over-restrictive. In order to alleviate this problem, we propose constructing a confidence map to indicate how likely the estimated flow vectors in the above step are correct, respectively based on the pixel-wise motion coherence and segment-wise model confidence. In what follows, we first detect image occlusion using the motion information.

Occlusion detection. To compute the occlusion in I_1, we simply warp I_2 to I_1 based on the motion vectors in the flow field of I_2. If a pixel $\mathbf{x}$ in I_1 does not receive projection from I_2, we set its occlusion value $O(\mathbf{x})$ to 1; otherwise, $O(\mathbf{x})$ is set to 0. We do not use a global optimization to enforce smoothness since it is found that the detected map is already sufficiently usable for the purpose of our motion confidence evaluation.

Pixel-wise motion coherence. In the following discussion, we denote by $(\mathbf{w}_1^0, \mathbf{w}_2^0)$ the flow pair estimated in the initial step and $(\mathbf{w}_1^s, \mathbf{w}_2^s)$ the flow pair computed in the second step for images I_1 and I_2 respectively. We construct function $E_p(\mathbf{w}_1^i, \mathbf{x})$ for each point $\mathbf{x}$ in flow field $\mathbf{w}_1^i$ to measure the motion coherence:

$$E_p(\mathbf{w}_1^i, \mathbf{x}) = \exp(-\frac{\sum_{c=1}^{3} |I_2(\mathbf{x} + \mathbf{w}_1^i(\mathbf{x}), c) - I_1(\mathbf{x}, c)|^2}{3\sigma_I^2}) \cdot \exp(-\frac{\|\mathbf{w}_1^i(\mathbf{x}) + \mathbf{w}_2^i(\mathbf{x} + \mathbf{w}_1^i(\mathbf{x}))\|^2}{\sigma_w^2}). \quad (8)$$

Here, the superscript $i = s$, denoting the confidence for the flow field in the 2nd step. $E_p(\mathbf{w}_1^s, \mathbf{x})$ is composed of two terms. $|I_2(\mathbf{x} + \mathbf{w}_1^s(\mathbf{x}), c) - I_1(\mathbf{x}, c)|^2$ models the color constancy between two matched pixels by a motion vector; $\|\mathbf{w}_1^i(\mathbf{x}) + \mathbf{w}_2^i(\mathbf{x} + \mathbf{w}_1^i(\mathbf{x}))\|^2$ models the motion coherence with respect to both images, similar to the cross check error defined in [3]. In all our experiments, $I_2(\mathbf{x} + \mathbf{w}_1^s(\mathbf{x}), c)$ and $\mathbf{w}_2^s(\mathbf{x} + \mathbf{w}_1^s(\mathbf{x}))$ are obtained using bilinear interpolation. The pixel-wise confidence for the flow computed in the second step is defined as:

$$C_p(\mathbf{w}_1^s, \mathbf{x}) = \begin{cases} \varsigma & \text{if O}(\mathbf{x})\text{=1,} \\ E_p(\mathbf{w}_1^s, \mathbf{x}) & \text{otherwise,} \end{cases} \quad (9)$$

where $O(\mathbf{x})$ is the occlusion value. ς is a constant to penalize the occluded pixels.

Segment-wise motion confidence. Only defining the pixel-wise motion coherence is not enough for the textureless segments with complex motion, since both the color constancy and motion coherence measure in E_p could have small values, which contradicts the true confidence definition. So we introduce the supplementary segment-wise confidence C_s, i.e. the confidence of the motion in a segment being affine, to handle the above problem. We define

$$C_s(\mathbf{w}_1^s, s) = \frac{\sum_{\mathbf{x} \in s} \exp(-\|\mathbf{w}_1^s(\mathbf{x}) - \mathbf{w}_1^0(\mathbf{x})\|^2 \; E_p(\mathbf{w}_1^0, \mathbf{x})/\sigma_A^2)(1 - O(\mathbf{x}))}{\sum_{\mathbf{x} \in s}(1 - O(\mathbf{x}))}, \quad (10)$$

where s denotes a segment. $(1 - O(\mathbf{x}))$ is to exclude the occluded pixels in computing the confidence of a segment since the initial flow is usually erroneous for these pixels. $E_p(\mathbf{w}_1^0, \mathbf{x})$ is defined in (8) by setting $i = 0$, modeling the pixel-wise flow confidence for the initial estimate. A small value of $E_p(\mathbf{w}_1^0, \mathbf{x})$ means we should not trust the initial flow. $\|\mathbf{w}_1^s(\mathbf{x}) - \mathbf{w}_1^0(\mathbf{x})\|^2$ measures how the flow in the second step is modified over that in initialization. If the initial flow is

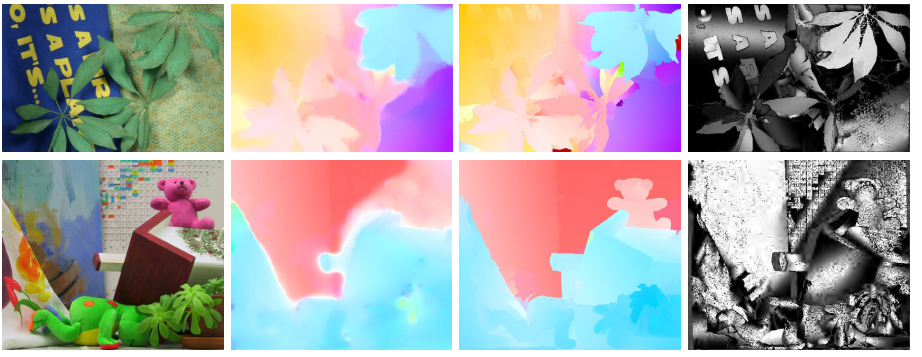


Fig. 6. Confidence maps for examples "Schefflera" (top row) and "Teddy" (bottom row). From left to right: one input image, initial optical flow (from step 1), optical flow with segmentation (from step 2), the constructed confidence map. Note that most pixels in the confidence map of "Teddy" are with high confidence. "Schefflera" contains non-rigid motion. So more flow vectors are problematic.

trustworthy, i.e., with large $E_p(\mathbf{w}_1^0, \mathbf{x})$, a large difference between $\mathbf{w}_1^s(\mathbf{x})$ and $\mathbf{w}_1^0(\mathbf{x})$ indicates that the affine model in a segment does not fit the actual pixel motion. So we should re-estimate the flow vectors for these pixels.

In (10), C_s returns a small value only if many pixels in a segment have high initial flow confidence and the corresponding initial flow vectors are quite different from those with segmentation (from step 2). It reflects that the parametric affine motion estimate in one segment is erroneous. The final confidence for $\mathbf{w}_1^s$ is a combination of the two measures:

$$\mathbf{conf}(\mathbf{x}) = C_p(\mathbf{w}_1^s, \mathbf{x}) \cdot C_s(\mathbf{w}_1^s, s(\mathbf{x})), \tag{11}$$

where $s(\mathbf{x})$ denotes the segment that contains pixel $\mathbf{x}$. Two examples of $\mathbf{conf}(\cdot)$ are shown in Fig. 6 where dark pixels indicate the possible erroneous estimates generated in the segmentation step (step 2). In these maps, many low confidence pixels are in non-rigid bodies.

3.5 Final Variational Model

We integrate the estimated confidence map into a final flow refinement phase to correct the possible flow errors due to segmentation. The final energy function is defined as

$$E_2(u,v) = \int_{\Omega_1} (1 - O(\mathbf{x})) \sum_{c=1}^{3} \Psi_D(|I_2(\mathbf{x}+\mathbf{w}, c) - I_1(\mathbf{x}, c)|^2) + \beta \mathbf{conf}(\mathbf{x}) \|\mathbf{w} - \mathbf{w}_1^s\|^2 + \alpha \Psi_S(\|\nabla u\|^2 + \|\nabla v\|^2) d\mathbf{x}. \tag{12}$$

There are three energy terms defined in $E_2(u,v)$. $(1 - O(\mathbf{x}))$ is to make color distance not be considered on the occluded pixels. $\beta \mathbf{conf}(\mathbf{x}) \|\mathbf{w} - \mathbf{w}_1^s\|^2$ imposes

a soft constraint. When weight $\mathbf{conf}(\mathbf{x})$ has a large value, the flow computed with segmentation (in step 2) will be trusted. Otherwise, other energy terms will be more influential in estimating the flow vector.

$E_2(u, v)$ selectively refines the flow vectors computed in the segmentation step, where the confidence map provides an essential clue whether the flow estimated in this step is correct or not. We minimize (12) by solving the corresponding Euler-Lagrange equations, similar to how we minimize (3).

4 Experimental Results

Experiments on both the synthesized and real images were conducted. Parameters in all experiments are configured with constant values as shown in table 1.

Table 1. Parameters used in our experiments

(a) stage 1 & 2

α	ϵ_D	ϵ_S
50	0.1	0.01

(b) stage 3

α	β	ϵ_D	ϵ_S	σ_I	σ_w	σ_A	ς
30	100	0.1	0.01	80	0.15	0.3	0.2

In regard to the parameter adjustment, specifically, ϵ_D and ϵ_S are set only for numerical stability. α and β are used to balance the energy terms. Note that the smoothness weight is set lower in our final stage as the segmentation information is incorporated. σ_I, σ_w and σ_A control the impact of the terms in constructing the confidence map. Larger values imply lower impact.

4.1 Quantitative Evaluation

Quantitative evaluation of the optical flow algorithm is conducted using the dataset in [23]. The overall rank of our method is high amongst all recorded optical flow algorithms on the Middlebury website based on the average angular error (AAE). We show in the second column of table 2 the average rank of the top eight algorithms at the moment we submit the data. Other columns on the right only show the AAE around motion discontinuities. Statistics show that our method has an advantage in faithfully preserving motion boundaries. The optical flow results are shown in Fig. 7.

For the computation speed, although the segmentation adds extra cost to the flow estimation, by using the multigrid scheme in the first and third steps, the total running time to process one image does not increase much. Typically, for an image with size 316×252, the running time of our algorithm is about 15 seconds on a PC with an Intel Core2Due 2.4G CPU.

Table 2. Average Angular Error (AAE) comparison obtained from the Middlebury website [23]. The second column shows the average ranks and other columns on the right show AAEs around the motion discontinuities.

Algorithm	Avg. Rank	Army	Mequon	Schefflera	Wooden	Grove	Urban	Yosemite	Teddy
Our Method	3.1	13.5_2	14.9_4	17.3_2	$\mathbf{18.1_1}$	$\mathbf{4.16_1}$	21.4_5	3.49_2	$\mathbf{9.23_1}$
Fusion	3.3	13.7_3	$\mathbf{8.91_1}$	$\mathbf{9.68_1}$	19.8_2	4.82_5	$\mathbf{17.3_1}$	5.78_8	13.6_4
SO prior	4.7	$\mathbf{11.2_1}$	13.1_2	17.7_3	20.9_3	5.27_7	22.0_7	6.88_9	13.4_3
Dynamic MRF	4.8	15.0_4	15.3_5	17.8_4	23.7_4	4.63_4	19.1_4	5.29_5	17.8_7
LP Registration	5.8	16.8_5	13.8_3	17.8_4	24.5_5	4.56_3	21.4_5	5.48_7	17.9_8
B. & A.	6.4	18.7_6	21.9_7	23.7_6	30.0_6	5.23_6	18.2_2	4.44_3	14.3_5
2D-CLG	6.6	22.6_9	16.9_6	28.2_9	31.1_8	4.25_2	22.2_8	$\mathbf{3.14_1}$	12.9_2
H. & S.	7.8	19.9_8	23.2_{10}	25.9_7	30.6_7	5.27_7	25.8_{10}	5.41_6	17.5_6

Fig. 7. Our optical flow estimates on a set of challenging examples. Incorporating segmentation and using paramatric motion models do not degrade our results because of the multi-step estimation framework.

Table 3. Average percentage of the mis-matched pixels (defined in Sect. 4.2) for the three sequences by warping frames 10, 15 and 20 to frame 0. For comparison, we also show the statistics from the method of Black and Anandan [7].

	Our results	Black and Anandan [7]
frame 10→0	0.88 %	3.33 %
frame 15→0	3.42 %	5.98 %
frame 20→0	6.13 %	9.26 %

(a) Our result (b) Black and Anandan [7]

Fig. 8. One example of warping frame 15 to frame 0

4.2 Image Warping Results

We also conducted experiments on natural image sequences to estimate the flow field. By warping a frame in the tail back to one at the head using the flow computed in all intermediate frames, we can evaluate the quality of optical flow in terms of the accumulated accuracy. Three different sequences are used[1] in experiments with the following evaluation criteria. We first compute the intensity difference between the warped frames and the original one. Then the average percentage of pixels whose intensity difference is greater than 0.1 of the maximum intensity is computed. The statistics are shown in Table 3, attained by warping frames 10, 15, 20 to frame 0, respectively.

One warping result is shown in Fig. 8 where we warp frame 15 back to frame 0. To produce the result in (b), we use the code from the authors by hand tuning the parameters. The close-ups in the bottom row show that our method faithfully preserves motion discontinuities and produces sharp warping boundaries.

[1] Available at `http://www.cse.cuhk.edu.hk/~leojia/publication.html`

5 Conclusion

In this paper, we have described a segmentation-embedded optical flow framework which can be used to compute accurate flow field as well as high quality motion boundary. The proposed method accommodates the parametric and segmented motion estimation in a variational model. Then a confidence map is constructed to measure the confidence whether the segmentation and the corresponding motion model suit the flow estimation or not. This map enables the recognition of non-rigid motion and detection of the error caused by segmentation. Evaluation on the Middelbury data set validated the effectiveness of our method. Our segmentation-based method produces sharp motion boundary, having a clear advantage in applications such as video editing.

Acknowledgements

We thank Guofeng Zhang for his comments on this paper. This work was fully supported by a grant from the Research Grants Council of Hong Kong (Project No. 412708).

References

1. Zitnick, C.L., Kang, S.B., Uyttendaele, M., Winder, S.A.J., Szeliski, R.: High-quality video view interpolation using a layered representation. ACM Trans. Graph 23(3), 600–608 (2004)
2. Sun, J., Li, Y., Kang, S.B.: Symmetric stereo matching for occlusion handling. In: CVPR, vol. 2, pp. 399–406 (2005)
3. Yang, Q., Wang, L., Yang, R., Stewénius, H., Nistér, D.: Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. In: CVPR, vol. 2, pp. 2347–2354 (2006)
4. Zitnick, C.L., Jojic, N., Kang, S.B.: Consistent segmentation for optical flow estimation. In: ICCV, pp. 1308–1315 (2005)
5. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI, pp. 674–679 (1981)
6. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artif. Intell. 17(1-3), 185–203 (1981)
7. Black, M.J., Anandan, P.: The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. Computer Vision and Image Understanding 63(1), 75–104 (1996)
8. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
9. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. International Journal of Computer Vision 61(3), 211–231 (2005)
10. Bruhn, A., Weickert, J.: Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In: ICCV, pp. 749–755 (2005)

11. Xiao, J., Cheng, H., Sawhney, H.S., Rao, C., Isnardi, M.A.: Bilateral filtering-based optical flow estimation with occlusion detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 211–224. Springer, Heidelberg (2006)
12. Ben-Ari, R., Sochen, N.: Variational stereo vision with sharp discontinuities and occlusion handling. In: ICCV (2007)
13. Tschumperlé, D., Deriche, R.: Diffusion tensor regularization with constraints preservation. In: CVPR (1), pp. 948–953 (2001)
14. Ju, S.X., Black, M.J., Jepson, A.D.: Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In: CVPR, pp. 307–314 (1996)
15. Black, M.J., Jepson, A.D.: Estimating optical flow in segmented images using variable-order parametric models with local deformations. IEEE Trans. Pattern Anal. Mach. Intell. 18(10), 972–986 (1996)
16. Mémin, É., Pérez, P.: Hierarchical estimation and segmentation of dense motion fields. International Journal of Computer Vision 46(2), 129–155 (2002)
17. Cremers, D., Schnörr, C.: Motion competition: Variational integration of motion segmentation and shape regularization. In: DAGM-Symposium, pp. 472–480 (2002)
18. Amiaz, T., Kiryati, N.: Piecewise-smooth dense optical flow via level sets. International Journal of Computer Vision 68(2), 111–124 (2006)
19. Black, M.J.: Combining intensity and motion for incremental segmentation and tracking over long image sequences. In: ECCV, pp. 485–493 (1992)
20. Khan, S., Shah, M.: Object based segmentation of video using color, motion and spatial information. In: CVPR, vol. 2, pp. 746–751 (2001)
21. Brox, T., Bruhn, A., Weickert, J.: Variational motion segmentation with level sets. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 471–483. Springer, Heidelberg (2006)
22. Zhang, G., Jia, J., Xiong, W., Wong, T.T., Heng, P.A., Bao, H.: Moving object extraction with a hand-held camera. In: ICCV (2007)
23. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. In: ICCV (2007)
24. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. 24(5), 603–619 (2002)
25. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24(6), 381–395 (1981)

Similarity Features for Facial Event Analysis

Peng Yang[1], Qingshan Liu[1,2], and Dimitris Metaxas[1]

[1] Rutgers University, Piscataway NJ 08854, USA
peyang@cs.rutgers.edu
[2] National Laboratory of Pattern Recognition, Chinese Academy of Sciences
Beijing, 100080, China

Abstract. Each facial event will give rise to complex facial appearance variation. In this paper, we propose similarity features to describe the facial appearance for video-based facial event analysis. Inspired by the kernel features, for each sample, we compare it with the reference set with a similarity function, and we take the log-weighted summarization of the similarities as its similarity feature. Due to the distinctness of the apex images of facial events, we use their cluster-centers as the references. In order to capture the temporal dynamics, we use the K-means algorithm to divide the similarity features into several clusters in temporal domain, and each cluster is modeled by a Gaussian distribution. Based on the Gaussian models, we further map the similarity features into dynamic binary patterns to handle the issue of time resolution, which embed the time-warping operation implicitly. The haar-like descriptor is used to extract the visual features of facial appearance, and Adaboost is performed to learn the final classifiers. Extensive experiments carried on the Cohn-Kanade database show the promising performance of the proposed method.

1 Introduction

Automatic facial event analysis is a hot topic in the communities of computer vision and pattern recognition in recent years due to its potential applications in human-computer interface, biometrics, multimedia, and so on. Lots of methods have been proposed [1] [2] [3], and these methods can be categorized into two classes: image based methods and video based methods. The image based methods take only the mug shots (mostly the apexes) of the expressions into account [4] [5] [6] [7]. However, a natural facial event is dynamic, which evolves over time from the onset, the apex, to the offset, including facial expression. The image based methods ignore such dynamic characteristics, so it is hard for them to obtain good performances in a real world setting. Psychology studies also demonstrated the insufficiency of the image based methods [8] [9]. The video based methods attempt to analyze the facial event in the spatio-temporal domain [10] [11] [12] [13] [14] [15], and extensive experiments showed that they were better than the image based methods.

However, how to extract and represent the dynamics of facial is a key issue to the video based methods. The popular one is based on motion analysis. In [10],

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 685–696, 2008.

Black and Yacoob used the parametric motion models to describe the local facial dynamics, and took the parameters of local motion models as dynamic features. Torre [16] used condensation to track the local appearance dynamics with the help of subspace representation. In [17], the dynamics are represented by tracking the points of Active Shape Model [18]. Although the motion based methods are much intuitive, they are sensitive to image noise. Manifold learning was also employed to explore the intrinsic subspace of the facial expression events. [19] used the Leipschitz embedding to build a facial expression manifold, and [20] used multilinear models to construct a non-linear manifold model. How to find the intrinsic dimensions of the manifold is still an open problem. In addition, due to diversity of subjects, it is hard to obtain an efficient and general manifold structure. Recently, the volume features attracted much attention in dynamic event analysis [21] [13] [22], which embed the spatio and temporal variation together. The idea of the volume features is to regard the video data as a 3D volume and to extract the features directly from the volume data. Guoying [13] designed the volume local binary patterns (LBP) in the spatio-temporal domain to capture the dynamics of facial events. In [23], we developed the ensemble of Haar-like features with coding scheme for expression recognition.

Time resolution is another issue for video based method, especially in real environment, because there are many factors to make the data varied in different time resolutions. For example, different cameras have different capture speed; different subjects have different paces for the same expression; even the same person will have different response in different situation. Therefore, in practical systems, some pre-defined time-warping processing should be demanded. However, most previous works did not take this into account including recent volume features [13] [23], and they assumed the training and the testing data must have the same length and the same speed rate, i.e., the same time resolution.

In this paper, we propose a new feature representation named similarity feature to address the above issues. It is well known that each facial event will give rise to complex appearance variation, and when it approximates to the apex, its discrimination become more distinct [24]. The kernel features achieved much success in describing the complex image variation [25], which are actually similarity representation against the training samples. Inspired by the kernel features, we measure each sample against the given references with a similarity function, and define the log-weighted summarization of the similarities as the feature to describe the complexity of facial appearance. The cluster-centers of the apex images are selected as the references. To capture the temporal dynamics, we perform the K-means clustering on the similarity features in the temporal domain, and each cluster is modeled as a Gaussian distribution. Based on the Gaussian models, we further map the similarity features into dynamic binary patterns to handle the issue of the time resolution, which involve the time-warping processing implicitly. The haar-like descriptor is used to extract the low-level visual features as in [23], and Adaboost learning is adopted to build the final classifier. Our experiments are conducted on the well-known Cohn-Kanade database, and the experimental results demonstrate that the proposed method has an

encouraging performance. The proposed feature representation is similar to the harr-like volume features [21] [22], but it can handle the data in various time resolution without any assumption. We will give detailed comparisons against the related work in the experiments.

The rest paper is organized as follows: We first give the definition of the similarity features in section 2, and describe how to map the similarity features into the dynamic binary patterns in Section 3. Section 4 addresses the classifier design, and the experiments are reported in Section 5, followed by conclusions.

2 Similarity Features

The kernel trick has attracted much attention, since the SVM achieved much success in the field of machine learning [26]. The kernel features are actually a kind of similarity representation, which are composed of the similarities between a given sample and all the training samples with a nonlinear kernel function, and they can describe the complex variations of images efficiently [25]. As we knew, each facial event is behaved by complex facial appearance variations. Inspired by the kernel features, we develop the similarity features to represent the complexity of facial appearance for facial event analysis. Different from the kernel features, we do not use all the training samples in computing similarities. We only take the apex images into account due to their distinctness. To avoid the influence of different subjects, we perform the K-Means clustering on the apex images to divide them into several clusters, and we take the cluster-centers as the references. The reference selection is also beneficial to computation cost reduction.

The similarity feature is calculated as follows: Given the references $\{r_i\}, i = 1, 2, ..., R$ and a given sample x, the similarities of x against the references are

$$S(x) = \{f(x, r_i\}, i = 1, 2, ..., R, \tag{1}$$

where $f(x, y)$ is the similarity function. In our experiments, we simply use the $L-2$ distance as the similarity function, $f(x, y) = \|x - y\|^2$. Now each sample is described by a R-dimensional similarity vector. Because the video data has both spatio and temporal information, it needs high computational cost if we directly use the above similarity vector. For example, if the number of the references is 100 and the video has 100 frames, then basically we need to do computation in a 10^4 dimensional space. To reduce the computational complexity, we convert the similarity vector into a log-weighted similarity as the final similarity feature,

$$F(x) = \sum_{i=1}^{R} \log(f(r_i, x)). \tag{2}$$

3 Dynamic Binary Coding

In practice, the video data we obtained often has different time resolution, so it is necessary to align the data into a same time scale by a pre-defined time-warping

operation. Most previous work did not discuss this issue and assumed the given data in the same time resolution including recent volume feature based methods. In this paper, we apply a coding scheme to handle this issue without any assumption in describing the dynamics of the similarity features.

Although facial event evolves over time from the onset, the apex, to the offset, we only take the process from the onset to the apex into account for simplicity, for this process is demonstrated to be enough for recognition in almost all the previous works. To describe the dynamics, we assume that the process from the onset to the apex is comprised of several intrinsic states (patterns) along the temporal domain. Correspondingly it means each kind of similarity feature can be divided into several patterns in temporal domain. In our experiments, we set the number of the intrinsic temporal patterns to 5 for all kinds of the similarity features. Without loss of generality, in the following we discuss how to build the five-level models for one event based on a similarity feature.

Given the training similarity feature set $F = \{F_i\}, i = 1, 2, ..., N$, where N is the number of the training samples, and each sample F_i has different resolution in temporal domain, $F_i = \{F_i^t\}$, where t is the index of frames. We perform the K-Means algorithm on the feature set F to divide it into five clusters in the temporal domain, and each cluster is modeled by a Gaussian distribution, $N^k\{\mu^k, \sigma^k\}$, $k = 1, 2, ..., 5$, where μ and *sigma* represent the mean and the variance respectively. We take these five Gaussian models as the temporal patterns of the feature F. In this paper, we will use a lot of similarity features to represent one facial event, so the temporal patterns models of one facial event are an ensemble of these Gaussian models as:

$$E = \begin{cases} N_1^1(\mu_1^1, \sigma_1^1), N_1^2(\mu_1^2, \sigma_1^2), ..., N_1^5(\mu_1^5, \sigma_1^5) \\ N_2^1(\mu_2^1, \sigma_2^1), N_2^2(\mu_2^2, \sigma_2^2)..., N_2^5(\mu_2^5, \sigma_2^5) \\ \qquad\qquad \vdots \\ N_M^1(\mu_M^1, \sigma_M^1), N_M^2(\mu_M^2, \sigma_M^2)..., N_M^5(\mu_M^5, \sigma_M^5), \end{cases} \tag{3}$$

where the subscript is the index of the similarity feature, and M is the number of the similarity features.

As mentioned in Section 1, each sequence may have different time resolution and different number of frames t due to various reasons. In order to handle this issue, we adopt the coding scheme to further convert the similarity features to the dynamic binary patterns. Given a feature sequence F_i with t frames,$\{F_i^t\}$, based the temporal pattern models described above, we can first map each F_i^t into a five-dimensional binary vector, i.e., $F_i^t \longrightarrow b_i^t = \{v_c\}$, where $c = 1, 2, ..., 5$. v_c is binary, and it is computed by the Bayesian rule as:

$$v_c = \begin{cases} 1 \; if \;\; c = \underset{k}{\operatorname{argmax}} P(F_i^t | N^k), k = 1, 2, ..., 5; \\ 0 \; otherwise, \end{cases} \tag{4}$$

where $P(F_i^t|N^k)$ means the probability of F_i^t given the corresponding Gaussian model $\{N^k\}$. We can see that there has only one element is 1 and the other four are 0 in the 5-dimensional binary feature b_i^t. It means each feature in a temporal

point only belongs to one temporal pattern. We map all the $\{F_i^t\}$ in a sequence into the five-dimensional binary feature vectors, and compute their histogram and do normalization as in [27],

$$\varphi(F_i) = \frac{\sum_{i=1}^{t} b_i^t}{t}, \tag{5}$$

where $\varphi(F_i)$ is always a five-dimensional vector whatever the t is. Thus, $\varphi(F_i)$ is independent of the time resolution. We call $\varphi(F_i)$ the dynamic binary pattern, and we use it to represent the sequence. As in [13], the binary pattern is transferred into the decimal value for the final classifier design.

4 Classifier Design

4.1 Haar-Like Appearance Descriptor

Facial event is behaved by facial appearance variations. Besides taking the gray or color values as appearance descriptor directly, there have three popular local descriptors: Gabor [27] descriptor, haar-like descriptor [28], and LBP descriptor [29]. In this paper, we use the haar-like descriptor due to its simplicity, for it has obtained a good performance for face detection and expression recognition [28] [23]. Compared to the Gabor and LBP descriptors, the haar-like descriptor only needs simple add or minus operations, so its computation cost is much lower. We perform the harr-like descriptor on each frame to extract the visual appearance features, and based on each haar-like feature, we can obtain its corresponding similarity feature. Then we convert the similarity feature to the dynamic binary pattern for learning classifier. Figure 1 shows an example how to calculate the dynamic binary pattern for a haar-like descriptor.

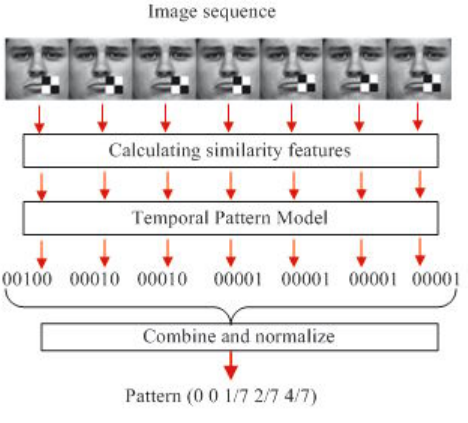


Fig. 1. Dynamic binary pattern calculation based on haar-like descriptor

4.2 Adaboost Learning

Since the number of the dynamic binary patterns is equal to the number of the haar-like features, each sequence has a large number of dynamic binary patterns. It is unrealistic to use all the dynamic binary patterns to design the classifier. Moreover, only parts of facial appearance are dominant in each facial event. Adaboost learning is a good tool to select some good features and combine them together to construct a strong classifier [28]. Therefore we adopt Adaboost to learn a set of discriminant dynamic binary patterns and use them to build the final classifier. In this paper, we take six basic facial expressions into account, i.e., happiness, sadness, angry, disgust, fear, and surprise, so it is a six-class recognition problem. We use the one-against-all strategy to decompose the six-class issue into multiple two-class issues. For each expression, we set its samples as the positive samples, and the samples of other expressions as the negative samples. Algorithm 1 summarizes the learning algorithm.

Algorithm 1. *Learning procedure*

1: Input the training image sequences (x_i, y_i),...,(x_n, y_n), $y_i \in \{+1, -1\}$.
2: Compute the similarity features for each image sequence.
3: Map the similarity features of each sequence into the dynamic binary patterns according to equation (4), and get $\varphi(F_{x_i})$.
4: Build one weak classifier on each $\varphi(F_{x_i})$.
5: Initialize weight $D_1(i) = 1/N$.
6: **for** $t = 1$ to T **do**
7: Find the classifier $h_t : \varphi(F_x) \rightarrow \{+1, -1\}$ that minimizes the error with respect to the distribution D_t. $h_t = \text{argmin}\, \varepsilon_j$, where $\varepsilon_j = \sum_{i=1}^{m} D_t(i)[y_i \neq h_j(\varphi(F_{x_i}))]$
8: Prerequisite: $\varepsilon_t < 0.5$, otherwise stop.
9: Choose $\alpha_t \in \mathbf{R}$, typically $\alpha_t = \frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}$ where ε_t is the weighted error rate of classifier h_t.
10: Update: $D_{t+1}(i) = \frac{D_t(i)\, e^{-\alpha_t y_i h_t(\varphi(F_{x_i}))}}{Z_t}$, where Z_t is a normalization factor.
11: **end for**
12: Output the final classifier: $H(x) = \text{sign} \sum_{t=1}^{T} \alpha_t h_t(\varphi(F_x))$

5 Experiments

Our experiments are conducted on the Cohn-Kanade facial expression database [30], which is widely used to evaluate the facial expression recognition algorithms. This database consists of 100 students aged from 18 to 30 years old, of which 65% are female, 15% are African-American, and 3% are Asian or Latino. Subjects are instructed to perform a series of 23 facial displays, six of which are prototypic emotions mentioned above. For our experiments, we select 300 image sequences from 96 subjects. The selection criterion is that a sequence is labeled as one of the six basic emotions. We randomly select 60 subjects as the

training set, and the rest of subjects is for the testing set. The face is detected automatically by Viola's face detector [28], and it is normalized to 64×64 as in [14] based on the location of the eyes.

The proposed work is related to the haar-like volume features [22], so we first compare our work with it. For simplicity, we denote our method as the DSBP and the haar-like volume features as the 3D haar. We also investigate the robustness of the proposed method, if the training samples and the testing samples have different length and different time resolution. ROC curve is used as the measurement tool to evaluate the performance, because it is more general and reliable than the recognition rate. The number of references is set to 5 for all the haar-like features in all the experiments.

5.1 Comparison to 3D Haar-Like Features

In this subsection, we compare the DSBP with the 3D haar. As mentioned above, the 3D haar takes the video data as the 3D volume data, and performs the haar-like descriptors in the spatio-temporal domain directly on the volume data, so it needs all the input sequence with same time resolution, i.e., the data has the same length and the same motion speed. The DSBP does not make such assumption, for it embeds the time-warping process in the dynamic binary coding. For fair comparison, we compare them under the same framework, and the training samples and the testing samples have the same length, but we make the data with different time resolution. Since the sequences in the Cohn-Kanade facial database have different lengthes, we use a fixed-length window to slide over the sequences to produce the fix-length samples.

We fix the training samples with 7 frames and 9 frames respectively. Figure 2 reports the ROC curves of the comparison experiment, and table 1 reports the area below the ROC curves. We can see that the performance of the DSBP is better than that of the 3D haar. This because: 1) similarity features are able to efficiently describe complex facial appearance; 2) the dynamic binary patterns are encoded based on the statistics and the Bayesian rule, so it is robust to some noise; 3) the samples generated

from the fix-length window should have different active speeds, but the DSBP is insensitive to active speeds.

Table 1. The Area under the ROC curves (the 3D haar and the DSBP)

Expression	9(xxxxxxxxx) frames		7(xxxxxxx) frames	
	3D Haar	DSBP	3D Haar	DSBP
Angry	0.934	0.957	0.893	0.935
Disgust	0.822	0.941	0.769	0.952
Fear	0.697	0.935	0.830	0.952
Happiness	0.977	0.997	0.978	0.997
Sadness	0.758	0.963	0.875	0.917
Surprise	0.974	0.999	0.982	0.999

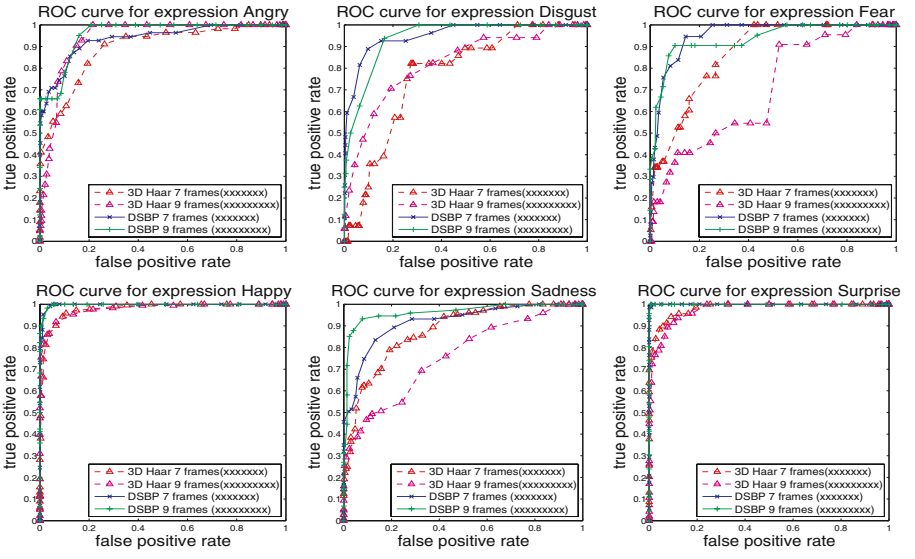


Fig. 2. ROC curves of six expressions in table 1

5.2 Robustness Analysis

The DSBP has another advantage against the 3D-haar: it has no requirement on the length of the samples. In the following, we will analyze its robustness if the training samples and the testing samples have different lengthes. We use sampling strategy to simulate this case. In following, the xxx0x0x means that we sample 5 frames from a sequence of 7 frames, where 0 means the corresponding frame is lost.

We first fix the training samples with the same length, but the length of the testing samples is variable. First, we fix the training samples with 7 frames, and the testing samples are with different number of frames. Table 2 reports a group

Table 2. The Area under the ROC curves (Training on 7(xxxxxxx) frames)

	Angry	Disgust	Fear	Happiness	Sadness	Surprisee
xxx00xxxx0xx	0.9758	0.9283	0.9623	0.9992	0.9455	1.0000
x0xxx0xxxxx0	0.9756	0.9407	0.9768	0.9991	0.9412	1.0000
xxxxxxxxxx	0.9313	0.9374	0.9401	0.9954	0.9086	0.9998
xxxxxxxxx	0.9314	0.9610	0.9471	0.9948	0.9185	0.9993
x000xxx	0.9277	0.9356	0.9537	0.9966	0.9223	0.9995
xx0xxxx	0.9499	0.9509	0.9366	0.9957	0.9136	1.0000
x0x0xxx	0.9369	0.9494	0.9486	0.9971	0.9204	0.9992
0xx0x	0.9422	0.9455	0.9503	0.9961	0.9167	0.9999
mean	0.9463	0.9436	0.9519	0.9968	0.9233	0.9997
standard variance	0.0194	0.0103	0.0128	0.0016	0.0131	0.0003

Table 3. The Area under the ROC curves (Training on 9(xxxxxxxxx) frames)

	Angry	Disgust	Fear	Happiness	Sadness	Surprise
xxx00xxxx0xx	0.9809	0.9361	0.9677	0.9985	0.9825	1.0000
x0xxx0xxxxx0	0.9780	0.9378	0.9706	0.9991	0.9713	1.0000
xxxxxxxxxx	0.9344	0.9053	0.8848	0.9970	0.9528	0.9988
xxxxxxxxx	0.9374	0.9392	0.9004	0.9951	0.9502	0.9985
x000xxx	0.9306	0.9129	0.9206	0.9969	0.9548	0.9991
xx0xxxx	0.9454	0.9217	0.9062	0.9970	0.9530	0.9991
x0x0xxx	0.9401	0.9218	0.9167	0.9969	0.9536	0.9990
0xx0x	0.9451	0.9048	0.9285	0.9965	0.9531	0.9996
mean	0.9490	0.9224	0.9244	0.9971	0.9589	0.9993
standard variance	0.0195	0.0141	0.0306	0.0012	0.0116	0.0006

of experimental results, where the length of testing samples from 12 to 5 and the sampling ratio is variant, the corresponding ROC curves are shown in 3. We can see that the DSBP is basically not influenced by the length variation of the testing data. We extend the length of the training samples to 9 frames, and use the same testing samples. Table 3 shows the experiment results and the corresponding ROC curves are displayed in 4. We can see that results are similar to those in Table 2 and 3. It means the DSBP is insensitive to the length variance and resolution variance of the testing samples. The large window size has a little better performance, because the large window captures much dynamics of the expressions.

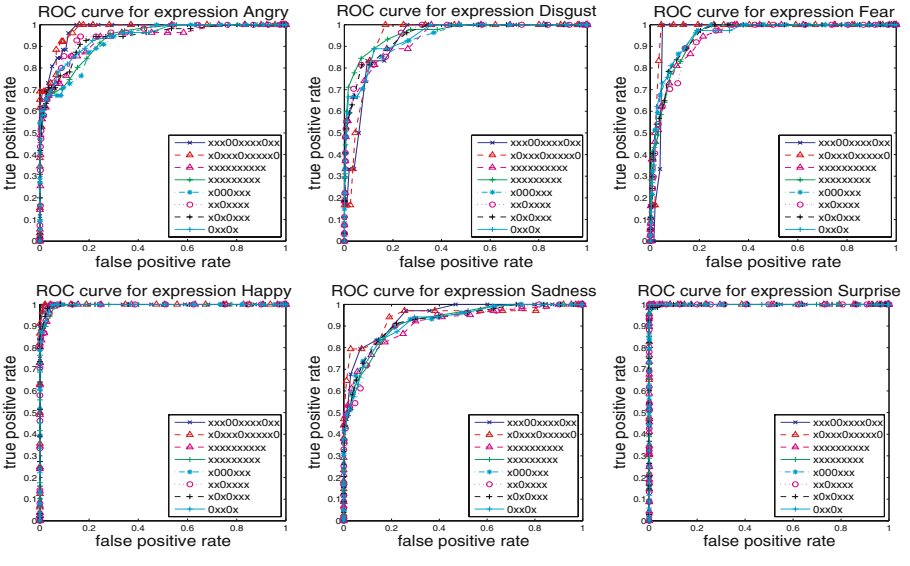


Fig. 3. ROC curves of six expressions in table 2

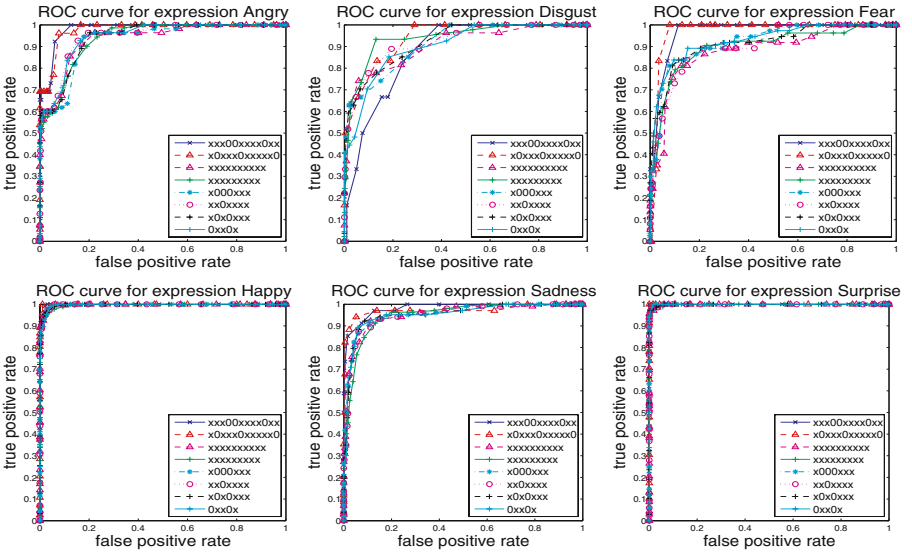


Fig. 4. ROC curves of six expressions in table 3

6 Conclusions

In this paper, we designed a novel similarity feature to describe the facial appearance for facial event analysis, which is inspired by the kernel features. The similarity feature is defined as the log-weighted summarization of the similarities between the given sample and the reference samples. We selected the references from the apexes of facial events due to their distinctness. In order to capture the dynamics of facial event, we divided the similarity features into several clusters in the temporal domain, and used the Gaussian distribution to model each cluster. Then we further mapped the similarity features into dynamic binary patterns to handle the issue of time-resolution, for this mapping processing involved the time-warping operation implicitly. The haar-like descriptor was used to extract the low-level visual features, and Adaboost was adopted to learn the final classifier. Experiments on the well-known Cohn-Kanade facial expression database showed the power of the propose method.

References

1. Fasel, B., Luettin, J.: Automatic Facial Expression Analysis: A Survey. Pattern Recognition 36, 259–275 (2003)
2. Pantic, M., Rothkrantz, L.J.M.: Automatic analysis of facial expressions: The state of the art. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 1424–1445 (2000)
3. Zeng, Z., Pantic, M., Roisman, G.I., Huang, T.S.: A survey of affect recognition methods: audio, visual and spontaneous expressions. In: Int. Conf. on Multimodal interfaces (2007)

4. Shan, C., Gong, S., McOwan, P.W.: Conditional mutual information based boosting for facial expression recognition. In: British Machine Vision Conference (2005)
5. Bartlett, M., Littlewort, G., Fasel, I., Movellan, J.: Real time face detection and facial expression recognition: Development and applications to human computer interaction. In: Computer Vision and Pattern Recognition Workshop on Human-Computer Interaction (2003)
6. Pantic, M., Rothkrantz, J.: Facial action recognition for facial expression analysis from static face images. IEEE Transactions on Systems, Man and Cybernetics (2004)
7. Shan, C., Gong, S., McOwan, P.W.: Robust facial expression recognition using local binary patterns. In: IEEE Int. Conf. on Image Processing (2005)
8. Bassili, J.: Emotion recognition: the role of facial movement and the relative importance of upper and lower areas of the face. J. Personality Socical Psychol. 37 (1979)
9. Ambadar, Z., Schooler, J., Cohn, J.F.: Deciphering the enigmatic face The importance of facial dynamics in interpreting subtle facial expression. Psychological Science (2005)
10. Black, M.J., Yacoob, Y.: Recognizing facial expressions in image sequences using local parameterized models of image motion. Int. J. Computer Vision 25, 23–48 (1997)
11. Yacoob, Y., Davis, L.: Computing spatio-temporal representations of human faces. Computer Vision and Pattern Recognition (1994)
12. Cohen, I., Sebe, N., Chen, L., Garg, A., Huang, T.: Facial expression recognition from video sequences Temporal and static modeling. Computer Vision and Image Understanding 91, 160–187 (2003)
13. Zhao, G., Pietikainen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE Trans. Pattern Anal. Mach. Intell. 29, 915–928 (2007)
14. Tian, Y.: Evaluation of face resolution for expression analysis. In: Computer Vision and Pattern Recognition Workshop on Face Processing in Video (2004)
15. Yeasin, M., Bullot, B., Sharma, R.: From facial expression to level of interest: A spatio-temporal approach. Computer Vision and Pattern Recognition (2004)
16. Torre, F., Yacoob, Y., Davis, L.: A probabilistic framework for rigid and non-rigid appearance based tracking and recognition. In: The Fourth IEEE Int. Conf. on Automatic Face and Gesture Recognition (2001)
17. Cohn, J.: Automated analysis of the configuration and timing of facial expression. What the face reveals (2nd edition): Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS), 388 – 392 (2005)
18. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models: their training and application. Comput. Vis. Image Underst. 61, 38–59 (1995)
19. Hu, C., Chang, Y., Feris, R., Turk, M.: Manifold based analysis of facial expression. In: Computer Vision and Pattern Recognition Workshop (2004)
20. Lee, C.S., Elgammal, A.: Facial expression analysis using nonlinear decomposable generative models. In: Zhao, W., Gong, S., Tang, X. (eds.) AMFG 2005. LNCS, vol. 3723, pp. 17–31. Springer, Heidelberg (2005)
21. Ke, Y., Sukthankar, R., Hebert, M.: Efficient visual event detection using volumetric features. In: IEEE International Conference on Computer Vision (2005)
22. Cui, X., Liu, Y., Shan, S., Chen, X., Gao, W.: 3d haar-like features for pedestrian detection. In: IEEE International Conference on Multimedia and Expo. (2007)

23. Yang, P., Liu, Q., Metaxas, D.N.: Boosting coded dynamic features for facial action units and facial expression recognition. Computer Vision and Pattern Recognition (2007)
24. Tversky, A.: Features of similarity. Psychological Review (1977)
25. Liu, Q., Jin, H., Tang, X., Lu, H., Ma, S.: A new extension of kernel feature and its application for visual recognition. Neurocomput. 71, 1850–1856 (2008)
26. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2, 121–167 (1998)
27. Daugman, J.: Demodulation by complex-valued wavelets for stochastic pattern recognition. Int'l J. Wavelets, Multiresolution and Information Processing (2003)
28. Viola, P., Jones, M.: Robust real-time object detection. Int. J. Computer Vision 57, 137–154
29. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. 24, 971–987 (2002)
30. Kanade, T., Cohn, J., Tian, Y.L.: Comprehensive database for facial expression analysis. In: Proceedings of the 4th IEEE Int. Conf. on Automatic Face and Gesture Recognition (FG 2000) (2000)

Building a Compact Relevant Sample Coverage for Relevance Feedback in Content-Based Image Retrieval

Bangpeng Yao[1], Haizhou Ai[1], and Shihong Lao[2]

[1] Computer Science & Technology Department, Tsinghua University, Beijing, China
[2] Sensing & Control Technology Laboratory, Omron Corporation, Kyoto, Japan

Abstract. Conventional approaches to relevance feedback in content-based image retrieval are based on the assumption that relevant images are physically close to the query image, or the query regions can be identified by a set of clustering centers. However, semantically related images are often scattered across the visual space. It is not always reliable that the refined query point or the clustering centers are capable of representing a complex query region.

In this work, we propose a novel relevance feedback approach which directly aims at extracting a set of samples to represent the query region, regardless of its underlying shape. The sample set extracted by our method is competent as well as compact for subsequent retrieval. Moreover, we integrate feature re-weighting in the process to estimate the importance of each image descriptor. Unlike most existing relevance feedback approaches in which all query points share a same feature weight distribution, our method re-weights the feature importance for each relevant image respectively, so that the representative and discriminative ability for all the images can be maximized. Experimental results on two databases show the effectiveness of our approach.

1 Introduction

Recently Content-Based Image Retrieval (CBIR) has been an active research topic. Typical CBIR systems use visual contents for image representation and similarity computation. Good surveys on CBIR can be found in [1] and [2].

One of the most challenging problems in CBIR is the "semantic-gap" problem. It means the low level features used to represent an image do not necessarily represent the human perception of that image. Techniques that were applied to reduce the semantic gap mainly include: (1) using object ontology to define high-level concepts [3]; (2) using machine learning methods to associate low-level features with query concepts [4,5]; (3) using relevance feedback (RF) [6,7,8,9,10,11] to learn users' intention. Compared with object ontology and machine learning, which mainly rely on offline learning, RF is an online approach and has been shown to be effective in boosting image retrieval accuracy [2]. During retrieval with RF, users interact with the system and give feedback scores to the images retrieved by the system. Based on the feedback, the system dynamically updates its query structure so that it can better capture users' semantic concepts.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 697–710, 2008.

1.1 Related Work

A typical RF approach is to identify the "ideal" query in user's mind. The classical query-point movement methods [6,7] represent the ideal query as a single point in feature space, and try to move this point toward relevant images as well as away from irrelevant images. Recently, query expansion methods [8,9] become more widely used as they can identify more complex query regions by using multiple queries. In these approaches, the relevant images are divided into many clusters, and the cluster centers are treated as new queries. But there are still two unsolved issues in the clustering based approaches: (1) these methods only used user-labeled relevant images, while neglecting the information contained in irrelevant images; (2) it lacks theoretical support that the modified query points are competent to represent the user's intention.

Another kind of RF approach is to find an appropriate transformation that maps the original feature space into a space that better models the user's high-level concepts. This is usually done by feature re-weighting to dynamically update the similarity metric. Techniques frequently used in re-weighting include Rocchio's formula [6] and machine learning. One problem in most of such approaches is that, only one similarity metric is obtained in each iteration, and this metric will be applied to all the query points. But in some applications different query points may require different feature weight distributions. An example of this situation in face image retrieval is shown in Fig. 1(b-e). In some clustering based RF methods [9] this problem is alleviated by using a specific similarity metric for each cluster center. However, these metrics are not optimal, because on the one hand, they depend on the quality of clustering; on the other hand, when learning these metrics, information in other clusters and irrelevant images are not used. In [12], a "local distance function" method was proposed which can obtain a distance function for each query image. But this approach treats all relevant images as queries. So applying such methods in RF will greatly improve the computational burden, and possibly fail to identify the "ideal" query region.

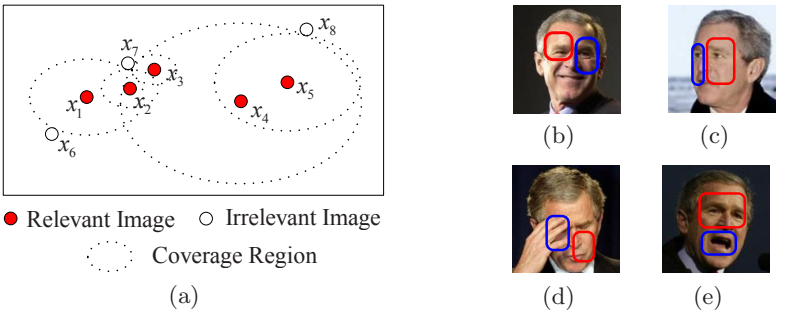


Fig. 1. (a) An illustration of sample coverage. $\{x_1, x_4\}$ is the minimum set that can cover all the other relevant images. (b)-(e) A same facial part may play different roles in similarity measure for face image retrieval. For example, intuitively the red regions should be more significant and discriminative than the blue ones.

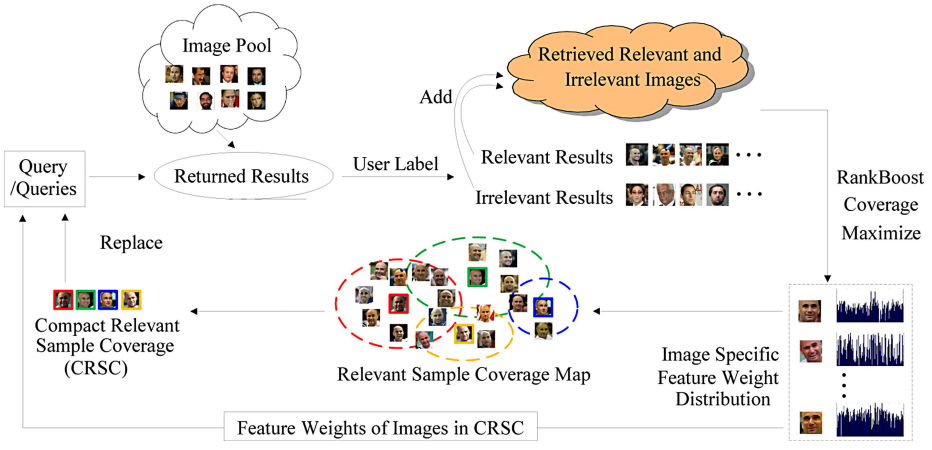


Fig. 2. Overview of our relevance feedback approach

1.2 Outline of Our Approach

In order to address the above issues, in this paper we propose a novel FR approach shown in Fig. 2. Our method directly aims to select a set of images that are competent as well as compact to represent the query region. In one RF iteration, the user specifies whether the retrieved images are relevant to the query, and assigns a relevance score to each relevant image. Based on the user's feedback, we define each relevant image's coverage set, which contains the set of relevant images that can be solved by this image in the nearest neighbor rule. Then, we use a boosting method to obtain a specific feature weight distribution for each relevant image respectively, so that the coverage set of each relevant image can be maximized.

After the coverage maximization stage, our method selects a minimum subset of images that can cover all the other relevant images. This set is called a Compact Relevant Sample Coverage (CRSC). We show that the CRSC extraction problem can be converted to the Minimum Dominating Set (MDS) [13] problem in graph theory. In this work we present an Improved Reverse Heuristic (IRH) method to solve this problem. Images in the CRSC with their feature weight distributions obtained in the coverage maximization stage will be used as new query points in the next retrieval step.

Major contributions in this paper are:

- Using CRSC to represent user's perception, instead of the clustering centers. All the information contained in user's feedback, including irrelevant images, relevant images and their relevance scores, can be used.
- The RankBoost method for simultaneous feature re-weighting and sample coverage maximization.
- The IRH solution for CRSC extraction.
- Each query point has a specific feature weight distribution, so that the representative and discriminative ability for each query can be better utilized.

The rest of this paper is organized as follows. In Sect. 2 we introduce some notations used in the paper and our similarity measure. Section 3 and 4 describe the coverage maximization and CRSC extraction approaches in detail. Experimental results are shown in Sect. 5. Finally in Sect. 6 is the conclusion.

2 Notations and Similarity Measure

In one RF iteration, we have a set of retrieved images $\boldsymbol{X}$. It contains N_{R} relevant images $\boldsymbol{X}_{\mathrm{R}} = \{x_{\mathrm{R},1}, \cdots, x_{\mathrm{R},N_{\mathrm{R}}}\}$ and N_{I} irrelevant images $\boldsymbol{X}_{\mathrm{I}} = \{x_{\mathrm{I},1}, \cdots, x_{\mathrm{I},N_{\mathrm{I}}}\}$. $x_{\mathrm{R},r}$'s relevance score is v_r. The CRSC of $\boldsymbol{X}$ is denoted as $\widetilde{\boldsymbol{X}} = \{\tilde{x}_1, \cdots, \tilde{x}_Q\}$.

Assume that each image is represented as a P dimensional vector. The distance from a relevant image $x_{\mathrm{R},r}$ to another image x is measured by[1]

$$\mathcal{D}(x_{\mathrm{R},r}, x) = \sum_{p=1}^{P} w_{r,p} d_p(x_{\mathrm{R},r}, x) \tag{1}$$

where $d_p(x_{\mathrm{R},r}, x)$ is the distance from $x_{\mathrm{R},r}$ to x measured by feature p, $w_{r,p}$ is feature p's weight when measuring the distance from $x_{\mathrm{R},r}$ to x. The image-specific feature weight distribution for $x_{\mathrm{R},r}$ is denoted as $W_r = \{w_{r,1}, \cdots, w_{r,P}\}$. Note that because we assign different feature weight distributions to different relevant images, the distance from $x_{\mathrm{R},r}$ to $x_{\mathrm{R},j}$ ($\mathcal{D}(x_{\mathrm{R},r}, x_{\mathrm{R},j})$) is not always equal to the distance from $x_{\mathrm{R},j}$ to $x_{\mathrm{R},r}$ ($\mathcal{D}(x_{\mathrm{R},j}, x_{\mathrm{R},r})$), because $w_{r,p} \neq w_{j,p}$.

In our approach, if $x_{\mathrm{R},r} \in \widetilde{\boldsymbol{X}}$, in the next iteration W_r will be used to measure the distance from $x_{\mathrm{R},r}$ to the images in the candidate pool. The feature weight distribution for $\tilde{x}_q$ is denoted as $\widetilde{W}_q = \{\tilde{w}_{q,1}, \cdots, \tilde{w}_{q,P}\}$. In the retrieval stage, the distance from the set of multiple query points in $\widetilde{\boldsymbol{X}}$ to an image x in the candidate pool is measured by an aggregate function,

$$\mathcal{D}_{agg}^{\tau}(\widetilde{\boldsymbol{X}}, x) = \frac{1}{Q}\sum_{q=1}^{Q}(\mathcal{D}(\tilde{x}_q, x))^{\tau} = \frac{1}{Q}\sum_{q=1}^{Q}\left(\sum_{p=1}^{P}\tilde{w}_{q,p} d_p(\tilde{x}_q, x)\right)^{\tau} \tag{2}$$

where a negative value of τ can make the smallest distance have the largest impact on the aggregate distance function [14]. We choose $\tau = -4$.

3 Coverage Maximization by Feature Re-weighting

3.1 Sample Coverage

Definition 1. *The Coverage Set of an image $x_{\mathrm{R},r}$ is defined as*

$$Cover(x_{\mathrm{R},r}) = \{x_{\mathrm{R},j} | x_{\mathrm{R},j} \in \boldsymbol{X}_{\mathrm{R}}, \mathcal{D}(x_{\mathrm{R},r}, x_{\mathrm{R},j}) < D\} \tag{3}$$

where $\mathcal{D}(x_{\mathrm{R},r}, x_{\mathrm{R},j})$ is the distance from $x_{\mathrm{R},r}$ to $x_{\mathrm{R},j}$, D is the distance from $x_{\mathrm{R},r}$ to its boundary image (the image in $\boldsymbol{X}_{\mathrm{I}}$ which is the nearest to $x_{\mathrm{R},r}$).

[1] In our method, we only need to measure the distance from a relevant image to another image (relevant or irrelevant images, or images in the candidate pool).

The definition is illustrated in Fig. 1(a). In the illustrated situation, x_8 is the boundary image of x_4 and x_5. According to the definition, we have: $Cover(x_1) = \{x_1, x_2\}$, $Cover(x_2) = \{x_2\}$, $Cover(x_3) = \{x_3\}$, $Cover(x_4) = \{x_2, x_3, x_4, x_5\}$, $Cover(x_5) = \{x_4, x_5\}$.

Sample coverage is a well-known concept in case-based reasoning [15]. From its definition, we can see that, the larger the coverage set of a sample, the more significant this sample, because it can correctly solve more relevant samples according to the nearest neighbor rule. In our RF problem, each relevant image has a relevance score. Therefore the coverage competence of a sample $x_{\mathrm{R},r}$ is measured by the sum of relevance scores of the images in its coverage, i.e.

$$\Psi_{Cover(x_{\mathrm{R},r})} = \sum_{x_{\mathrm{R},j} \in Cover(x_{\mathrm{R},r})} v_j. \tag{4}$$

From (3) and (4), we can see that, the definition and measurement of sample coverage makes use of all the information provided by the user. The irrelevant images serve to bound the coverage region, and the relevance scores are used to measure the competence of a coverage set.

3.2 The Image Specific Loss Function

The definition of sample coverage is based on a similarity measure. An image's coverage region can be modified by changing its feature weight distribution. Here, for each $x_{\mathrm{R},r}$, we learn a specific feature weight distribution $W_r = \{w_{r,1}, \cdots, w_{r,P}\}$ to maximize $\Psi_{Cover(x_{\mathrm{R},r})}$, the coverage ability of $x_{\mathrm{R},r}$.

We have two motivations to maximize the coverage ability of each relevant image. First, after the coverage ability of each sample is maximized, we can use a smaller number of images to cover all the relevant images. Second, we can obtain a specific feature weight distribution for each sample, which can be used in the subsequent retrieval stage.

According to Definition 1 and the concepts above, the loss function for obtaining W_r to maximize $x_{\mathrm{R},r}$'s coverage ability can be written as

$$\mathrm{Loss}_{W_r} = \sum_{j=1}^{N_{\mathrm{R}}} v_j \left\| \sum_{p=1}^{P} w_{r,p} d_p(x_{\mathrm{R},r}, x_{\mathrm{R},j}) > \min_{x_{\mathrm{I},i} \in \boldsymbol{X}_{\mathrm{I}}} \sum_{p=1}^{P} w_{r,p} d_p(x_{\mathrm{R},r}, x_{\mathrm{I},i}) \right\| \tag{5}$$

where $\|\mathrm{A}\|$ is an indicator function: if A is true $\|\mathrm{A}\| = 1$, otherwise $\|\mathrm{A}\| = 0$.

3.3 Loss Function Minimization Via RankBoost Learning

The algorithm to optimize (5) for all the relevant samples is shown in Fig. 3. Two parameters should be learned in order to minimize Loss_{W_r}: the weight distribution W_r and $x_{\mathrm{R},r}$'s boundary image (the irrelevant image which has the smallest distance to $x_{\mathrm{R},r}$). It is hard to learn the two parameters simultaneously, because the optimal weight distribution varies with respect to the selection of boundary image. Therefore, we treat each irrelevant image $x_{\mathrm{I},i}$ as $x_{\mathrm{R},r}$'s boundary respectively, and obtain a weight distribution $W_{r,i}$. $W_{r,i}$ can maximize the coverage

– **For** each relevant image $x_{\mathrm{R},r}$
  - **For** each irrelevant image $x_{\mathrm{I},i}$, treat it as $x_{\mathrm{R},r}$'s boundary sample.
    * The optimization objective becomes

$$W_{r,i} = \arg\min_{W_{r,i}} \sum_{j=1}^{N_{\mathrm{R}}} v_j \quad \sum_{p=1}^{P} w_{r,i,p} d_p(x_{\mathrm{R},r}, x_{\mathrm{R},j}) > \sum_{p=1}^{P} w_{r,i,p} d_p(x_{\mathrm{R},r}, x_{\mathrm{I},i}) \quad . \quad (6)$$

    * Decompose (6) into N_{R} ranked pairs:

$$\{(x_{\mathrm{R},r}, x_{\mathrm{R},1}), (x_{\mathrm{R},r}, x_{\mathrm{I},i})\}, \cdots, \{(x_{\mathrm{R},r}, x_{\mathrm{R},N_{\mathrm{R}}}), (x_{\mathrm{R},r}, x_{\mathrm{I},i})\}. \quad (7)$$

    * Assign initial importance value to all the ranked pairs. The importance for $\{(x_{\mathrm{R},r}, x_{\mathrm{R},j}), (x_{\mathrm{R},r}, x_{\mathrm{I},i})\}$ is $\frac{v_j}{\sum_{k=1}^{N_{\mathrm{R}}} v_k}$.
    * Treat each feature d_p as a weak ranker, run RankBoost P iterations to get a feature weight $w_{r,i,p}$ for each d_p.
    * Calculate $\ell_{r,i} = \sum_{j=1}^{N_{\mathrm{R}}} v_j \sum_{p=1}^{P} w_{r,i,p} d_p(x_{\mathrm{R},r}, x_{\mathrm{R},j}) > \sum_{p=1}^{P} w_{r,i,p} d_p(x_{\mathrm{R},r}, x_{\mathrm{I},i})$.
  - Find $i_r^* = \arg_i \min \ell_{r,i}$, and let $W_r = W_{r,i_r^*}$.

Fig. 3. Algorithm of sample coverage maximization and feature re-weighting

ability of $x_{\mathrm{R},r}$ when $x_{\mathrm{I},i}$ is the boundary, taking no account of the distance from $x_{\mathrm{R},r}$ to the other irrelevant images. The loss resulted from $W_{r,i}$ is $\ell_{r,i}$. After all the irrelevant images are considered, the minimum loss value ℓ_{r,i_r^*} is selected, and its associated W_{r,i_r^*} and x_{I,i_r^*} are the optimal feature weight distribution and boundary image respectively.

When $x_{\mathrm{I},i}$ is set as $x_{\mathrm{R},r}$'s boundary image, the optimization objective becomes (6). As shown in Fig. 3, here (6) is solved by RankBoost [16]. We start by decomposing (6) into a set of ranked pairs as shown in (7). We assume that the image features are P weak rankers, where $d_p(x_{\mathrm{R},r}, x_{\mathrm{I},i}) > d_p(x_{\mathrm{R},r}, x_{\mathrm{R},j})$ means $(x_{\mathrm{R},r}, x_{\mathrm{I},i})$ is ranked higher than $(x_{\mathrm{R},r}, x_{\mathrm{R},j})$ by the pth feature. Our goal is to find a strong ranker $\mathcal{D}$, which is a linear combination of $\{d_1, \cdots, d_P\}$ using a set of weights $\{w_{r,i,1}, \cdots, w_{r,i,P}\}$, so that $\mathcal{D}(x_{\mathrm{R},r}, x_{\mathrm{I},i})$ can be ranked higher than $\mathcal{D}(x_{\mathrm{R},r}, x_{\mathrm{R},j})$ for all the $j = 1, \cdots, N_{\mathrm{R}}$. The mis-ranking between $\mathcal{D}(x_{\mathrm{R},r}, x_{\mathrm{I},i})$ and $\mathcal{D}(x_{\mathrm{R},r}, x_{\mathrm{R},j})$ is penalized with v_j, the relevance score of $x_{\mathrm{R},j}$. This learning objective is exactly consistent with the formal ranking problem defined in [16], and thus the optimal $W_{r,i}$ can be found with RankBoost learning.

Like other boosting methods, RankBoost [16] operates iteratively and in each iteration, selects a "best" weak ranker and determines its importance. In the tth iteration, the best weak ranker h_t and its weight α_t is selected according to

$$h_t = \arg\max_{d_p} \sum_{j=1}^{N_{\mathrm{R}}} \rho_{t,j} \left(d_p(x_{\mathrm{R},r}, x_{\mathrm{I},i}) - d_p(x_{\mathrm{R},r}, x_{\mathrm{R},j}) \right) \quad (8)$$

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 + \sum_{j=1}^{N_{\mathrm{R}}} \rho_{t,j} (h_t(x_{\mathrm{R},r}, x_{\mathrm{I},i}) - h_t(x_{\mathrm{R},r}, x_{\mathrm{R},j}))}{1 - \sum_{j=1}^{N_{\mathrm{R}}} \rho_{t,j} (h_t(x_{\mathrm{R},r}, x_{\mathrm{I},i}) - h_t(x_{\mathrm{R},r}, x_{\mathrm{R},j}))} \right) \quad (9)$$

where $\rho_{t,j}$ is the importance of distance pair $\{(x_{\mathrm{R},r}, x_{\mathrm{R},j}), (x_{\mathrm{R},r}, x_{\mathrm{I},i})\}$ in the tth iteration. As shown in Fig. 3, $\{(x_{\mathrm{R},r}, x_{\mathrm{R},j}), (x_{\mathrm{R},r}, x_{\mathrm{I},i})\}$'s initial importance is $\rho_{1,j} = \frac{v_j}{\sum_{k=1}^{N_\mathrm{R}} v_k}$. After h_t is selected, the distance pair's importance value is updated by

$$\rho_{t+1,j} = \frac{\rho_{t,j} \exp(\alpha_t(h_t(x_{\mathrm{R},r}, x_{\mathrm{R},j}) - h_t(x_{\mathrm{R},r}, x_{\mathrm{I},i})))}{Z_t} \tag{10}$$

where Z_t is a normalization factor so that $\rho_{t+1,j}$ is a distribution.

In our method, once a feature has been chosen, it cannot be selected again. For each learning task we implement RankBoost P iterations, and thus each feature can have a weight value.

4 An Improved Reverse Heuristic Solution for CRSC Extraction

After all the relevant images' coverage sets are maximized, we shall extract the CRSC from $\boldsymbol{X}_\mathrm{R}$. Here we show that the CRSC extraction problem can be converted to the Minimum Dominating Set (MDS) problem [13] in graph theory. We propose an Improved Reverse Heuristic (IRH) method to solve this problem.

4.1 Convert CRSC Extraction to the MDS Problem

Definition 2. *The Dominating Set (DS) of a graph G is defined as,*

$\boldsymbol{S}$ is a subset of the vertex set $\boldsymbol{U}(G)$. $\boldsymbol{N}_G[\boldsymbol{S}]$ is the set of vertices in G which are in $\boldsymbol{S}$ or adjacent to a vertex in $\boldsymbol{S}$. If $\boldsymbol{N}_G[\boldsymbol{S}] = \boldsymbol{U}(G)$, then $\boldsymbol{S}$ is said to be a dominating set (of vertices in G).

If there does not exist another dominating set $\boldsymbol{S}'$ whose $|\boldsymbol{S}'| < |\boldsymbol{S}|$, then $\boldsymbol{S}$ is the Minimum DS (MDS) of G. ($|\boldsymbol{S}|$ is the number of vertices in $\boldsymbol{S}$).

Proposition 1. *CRSC extraction can be converted to the MDS problem.*

Proof. Given a set of user-labeled images $\boldsymbol{X}$, $\widetilde{\boldsymbol{X}} = \{\tilde{x}_1, \cdots, \tilde{x}_Q\}$ is its CRSC.

Build a directed graph G with N_R vertices, where N_R is the number of relevant images in $\boldsymbol{X}$. The vertex u_r corresponds to the relevant image $x_{\mathrm{R},r}$ in $\boldsymbol{X}$. In G, there is an edge from u_r to u_j iff $x_{\mathrm{R},r} \in \widetilde{\boldsymbol{X}}$ and $x_{\mathrm{R},j} \in Cover(x_{\mathrm{R},r})$.

According to G's construction process and Definition 2, $\boldsymbol{S} = \{s_1, \cdots, s_Q\}$ is the DS of G, where s_q is the point corresponds to $\tilde{x}_q$.

If G has another DS whose sample size is smaller than Q, then the corresponding images of this set is also a CRSC of $\boldsymbol{X}$. This contradicts the pre-condition that $\widetilde{\boldsymbol{X}}$ is the CRSC of $\boldsymbol{X}$. Therefore, $\boldsymbol{S}$ is the MDS of G.

Thus, if we find $\boldsymbol{S}$ from G, we find $\widetilde{\boldsymbol{X}}$ from $\boldsymbol{X}$. Therefore, extracting the CRSC from $\boldsymbol{X}$ can be converted to a MDS problem. □

The corresponding graph of the situation in Fig. 1(a) is shown in Fig. 4.

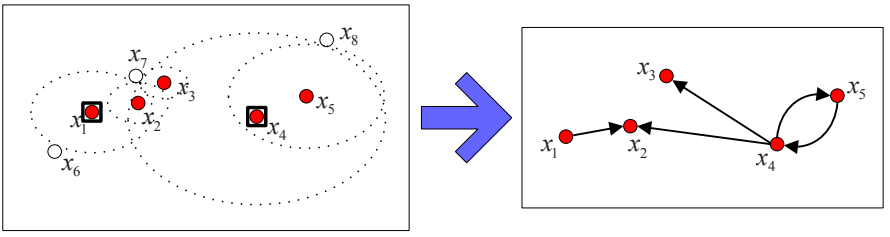


Fig. 4. The left figure is a sample coverage figure (the same as Fig. 1(a)), and the right is its corresponding graph. Extracting a Compact Relevant Sample Coverage of the left figure can be converted to finding the right figure's MSD, which is $\{x_1, x_4\}$.

4.2 The Improved Reverse Heuristic Solution

Based on the above observation, a method for MDS can be directly used to extract CRSC. However, MDS is a well-known NPC problem in graph theory. It is rather time-consuming to find a globally optimal solution. For approximate optimal solutions there are two well-known approaches, *Tabu Search (TS)* and *Reverse Heuristic Algorithm (RH)*. TS works by modifying an initial solution iteratively according to some searching criterions. Although in many situations TS can alleviate the local minima problem that exists in many other approaches, it relies too much on the initial solution, and its convergence speed might be very slow. RH is another algorithm to find approximate solutions for NP problems. In the MDS application, in each iteration of RH, the vertex that has the largest coverage will be selected, and this vertex and those that are connected with it will be removed from the graph. This approach can find an approximate dominating set rapidly. The drawback of RH is the local minima problem.

One reason for the local minima problem in heuristic based algorithms is that, the heuristic rule is not good enough. In the original RH algorithm, only sample coverage is considered. That is, the larger the coverage set, the more significant the sample. However, besides sample coverage, another concept, sample reachability is also important for measuring a sample's competence.

Definition 3. *The Reachability Set of a sample* $x_{\mathrm{R},r}$ *is defined as*

$$Reach(x_{\mathrm{R},r}) = \{x_{\mathrm{R},j} | x_{\mathrm{R},j} \in \boldsymbol{X}_{\mathrm{R}}, x_{\mathrm{R},r} \in Cover(x_{\mathrm{R},j})\}. \tag{11}$$

For a sample, the larger its reachability set, the less important this sample, because it can be covered by many other samples. Since both sample coverage and sample reachability reflect the importance of a sample, they should be combined to result in a more reliable measure. In our work a sample $x_{\mathrm{R},r}$'s competence for RH is measured as

$$\begin{aligned} Comp(x_{\mathrm{R},r}) &= \Psi_{Cover(x_{\mathrm{R},r})} - \Psi_{Reach(x_{\mathrm{R},r})} \\ &= \sum_{x_{\mathrm{R},j} \in Cover(x_{\mathrm{R},r})} v_j - \sum_{x_{\mathrm{R},k} \in Reach(x_{\mathrm{R},r})} v_k \end{aligned} \tag{12}$$

- **Input**: A set of relevant images $\boldsymbol{X}_{\mathrm{R}}$ and irrelevant images $\boldsymbol{X}_{\mathrm{I}}$.
- **Initialize**: $\boldsymbol{X} = NULL$.
- **While** $\boldsymbol{X}_{\mathrm{R}} \neq NULL$
  - For each $x_{\mathrm{R},r} \in \boldsymbol{X}_{\mathrm{R}}$, calculate its $Cover(x_{\mathrm{R},r})$ and $Reach(x_{\mathrm{R},r})$;
  - Get $x^*_{\mathrm{R}} \in \boldsymbol{X}_{\mathrm{R}}$ so that, $x^*_{\mathrm{R}} = \arg \max_{x_{\mathrm{R},r} \in \boldsymbol{X}_{\mathrm{R}}} Comp(x_{\mathrm{R},r})$; Ties are broken by selecting the sample with larger coverage set.
  - Append x^*_{R} to $\boldsymbol{X}$;
  - Remove $Cover(x^*_{\mathrm{R}})$ from $\boldsymbol{X}_{\mathrm{R}}$.
- **Output**: $\boldsymbol{X}$.

Fig. 5. Improved Reverse Heuristic Algorithm for Minimum Dominating Set Detection

Using (12), we propose an improved reverse heuristic solution for CRSC, shown in Fig. 5. The only difference between our method and the original RH is the heuristic rule.

5 Experiment

5.1 Databases and Evaluation Settings

In this section, we compare our method with some state-of-the-art methods, including Query Movement (Mindreader [7]), Query Expansion (Qcluster [9]), and pure Machine Learning (SVM with triangular kernel [5]).

Experiments are conducted on two publicly available databases: Labeled Faces in the Wild (LFW) [17] and Caltech 101 [18]. LFW is a database of face photographs designed for studying the problem of unconstrained face recognition. It contains 13,233 face images of 5,749 people collected from web, among which 57 people have more than 20 images. The LFW database contains large variations of head pose, illumination, and expression. The Caltech 101 database is a collection of object pictures belonging to 101 categories, such as airplane, panda, etc. Each category has 40 to 800 images. Most categories have about 50 images. The size of each image is roughly 300×200 pixels.

We use the two databases because face/object recognition and retrieval are both active research topics. Moreover both LFW and Caltech 101 are suitable for CBIR performance evaluation, because of their large size, great homogeneous and heterogeneous variations, and human annotated ground truth available.

5.2 Performance Measure and Initial Queries

We use two performance measures: Recall (Re) and Rank (Ra) to evaluate the effectiveness of these RF methods. Recall is defined as the number of retrieved relevant images over the total number of relevant images in the candidate pool. For two RF algorithms A and B, if $Re_A > Re_B$, then A is better than B in terms of Re, because A retrieves more relevant images than B. Rank is the average rank of the retrieved relevant images returned by the system. Obviously, if $Ra_A < Ra_B$, then A is better than B in terms of Ra.

Scope (Sc) is the number of images returned to the user in each RF iteration. It may also affect the performance of a RF approach. Here we measure the effectiveness of these RF methods when Sc is 40 and 80 respectively. The iteration number of each experiment is 4. Note that we only label "relevant" ($v_r = 1$) and "irrelevant" to the returned images, without providing different relevance scores. This is because in the databases human annotated ground truth is available.

It is known that the quality of initial query is important for CBIR. Having a frontal neutral face as the initial query usually achieves better retrieval results than that obtained by using a profile image. Thus, a system that performs well on selected queries does not necessarily work well on not-selected images. In this work, the initial queries are selected randomly. On LFW, we randomly selected 100 images from the 57 people with more than 20 images as initial queries. On Caltech 101, one random image per class was selected. On the two databases, average retrieval results are reported for performance evaluation.

5.3 Visual Features

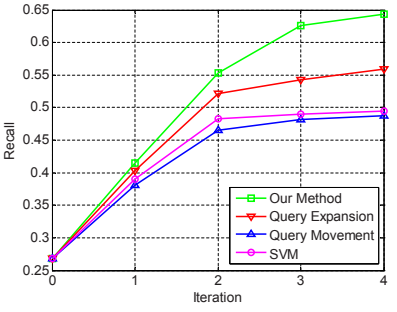
Early CBIR systems mainly rely on low-level features such as color and edge. With the advances of computer vision research in these years, many novel and domain-specific features are available. Here we use Local Binary Pattern (LBP) [19] for face similarity measure, and use Bag of Words (BoW) [18] to measure generic object similarity. Both LBP and BoW are recently developed features and have been widely used in face and object recognition respectively.

In the LBP representation, each face image is normalized to the size of 64×64 pixels according to two-eye centers. The normalized image is divided into 8×8 sub-regions. LBP histogram in each region is treated as a feature. The histogram similarity is measured using Chi-square distance, as in [19]. BoW is also a histogram based feature. We generate the texture codebook using 200 images that are randomly selected from the Caltech 101 database. The codebook histogram size is 10×10, and has 128 bins. In the BoW representation each histogram bin is treated as a feature.

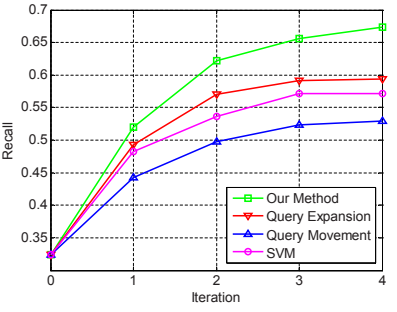
It should be noted that the selection of visual features in our experiment is not optimal. For example, we did not use any color information in extracting LBP and BoW features. It is possible that using other features can obtain better results than that reported in this paper. The focus of this paper is the RF scheme. Which feature is the best for CBIR is beyond the content of this paper.

5.4 Results and Analysis

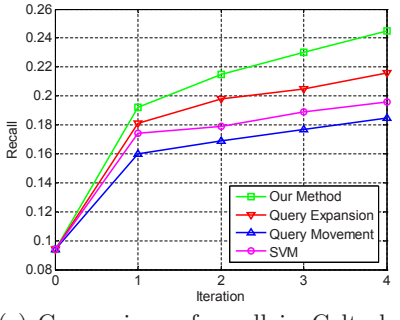
Experiment results of the four approaches are shown in Fig. 6 and Table 1. Figure 6 illustrates that, our method consistently outperforms the other approaches by a large margin. What is more, the margin increases with the iteration number. This is important for a RF system, because the images retrieved after 2 or 3 iterations are usually distinct from the initial query in terms of feature representation. However, these images are closely related to the query in human's perception. So we can say our method better reduces the "semantic gap" between feature representation and human perception.



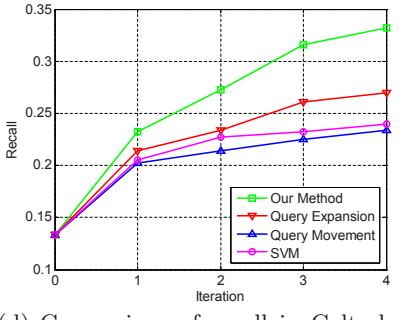
(a) Comparison of recall in LFW database when $Sc = 40$.



(b) Comparison of recall in LFW database when $Sc = 80$.



(c) Comparison of recall in Caltech 101 database when $Sc = 40$.



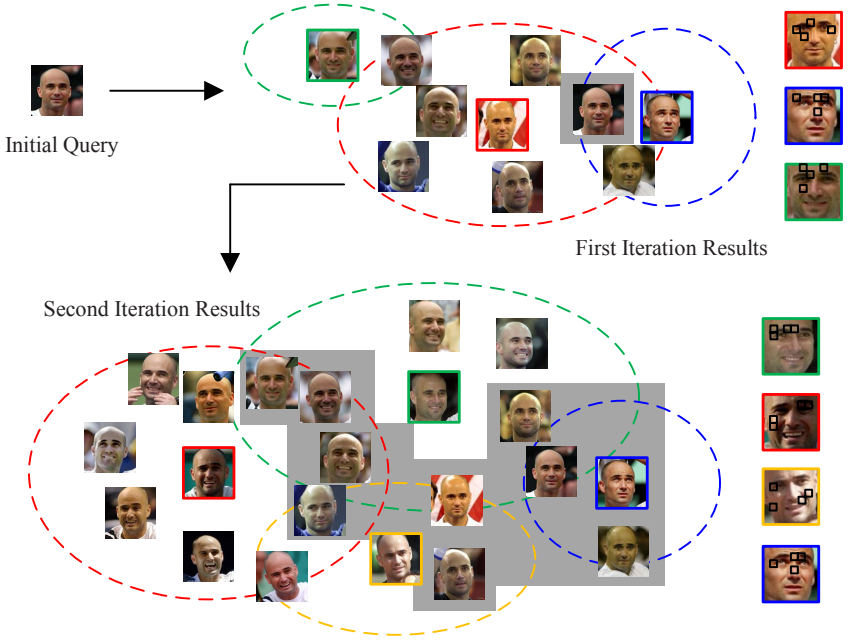
(d) Comparison of recall in Caltech 101 database when $Sc = 80$.

Fig. 6. Comparison of the four RF approaches on LFW and Caltech 101 databases

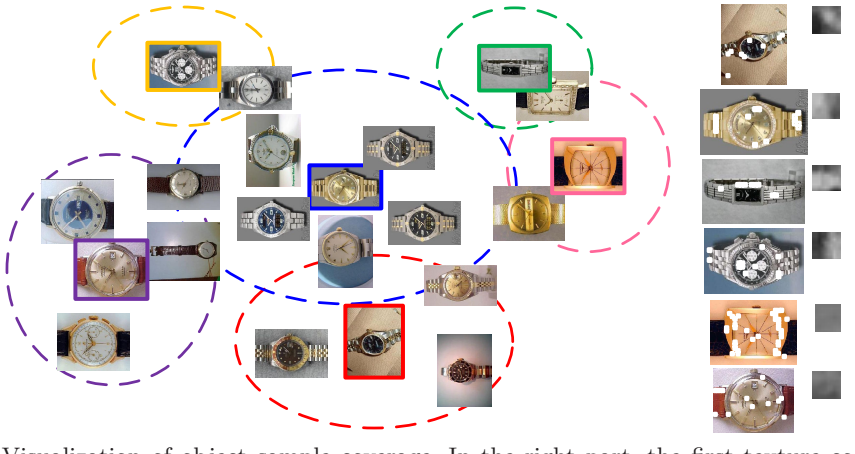
Table 1. Using Rank to evaluate different RF approaches' performance on LFW and Caltech 101 databases. The smallest rank in each experiment are in bold.

		$Sc = 40$					$Sc = 80$				
		It.=0	It.=1	It.=2	It.=3	It.=4	It.=0	It.=1	It.=2	It.=3	It.=4
LFW	Our Method	11.0	**10.3**	**11.6**	11.2	**16.1**	20.3	**18.5**	**18.8**	**14.4**	20.2
	QCluster	11.0	13.3	13.8	12.4	17.0	20.3	19.8	19.6	17.2	**15.9**
	Mindreader	11.0	11.7	12.6	11.9	20.3	20.3	20.6	19.2	18.1	18.4
	SVM	11.0	12.1	11.9	**10.7**	16.9	20.3	20.7	19.1	16.9	19.3
Caltech101	Our Method	13.2	**13.0**	14.2	**13.7**	16.3	30.6	28.8	**28.1**	**25.6**	**25.0**
	QCluster	13.2	15.3	14.9	15.2	**15.9**	30.6	**28.6**	29.0	29.1	27.7
	Mindreader	13.2	13.9	**13.3**	14.1	16.4	30.6	29.1	28.8	27.3	27.8
	SVM	13.2	14.7	15.0	15.3	16.9	30.6	30.3	29.8	28.9	27.4

One reason for our method's better performance in higher iterations is that, it uses the information in irrelevant images. With the increasing of iteration number, more and more irrelevant images are available as boundary image candidates. Thus the CRSC built by our method are more tight and more consistent with the human's perception. In our experiment, we find that when the iteration



(a) Visualization of face sample coverage. In the right part, the first four regions selected by RankBoost for the images in the CRSC are illustrated. We showed the retrieval process with 2 RF iterations, with one image of Agassi as the initial query. The images with shadow background means it has been retrieved in previous iterations.



(b) Visualization of object sample coverage. In the right part, the first texture code selected by RankBoost, and the position of these codes appear in the CRSC images are illustrated. Due to space limitation, we only show the final results with 2 RF iterations.

Fig. 7. Visualization of sample coverage. Images with colored frames form the Compact Relevant Sample Coverage in each figure. The dotted ellipses are coverage regions.

number is large, the number of images in the CRSC is usually larger than the number of clusters in the query expansion method. That is to say, the cluster centers are not competent to represent the whole query region.

Although SVM also uses irrelevant images, in the training procedure, it is likely that many relevant images are very similar to the initial query, and thus the classification boundary learned by SVM will be specialized to the initial query rather than human perception. This problem will not happen in our method, because the similar images are likely to be in a same coverage region, and only the images in CRSC will be used as new queries in the next iteration.

Furthermore, in most situations, images retrieved by our method have the smallest rank, as shown in Table 1. Actually, rank and recall are two contrary measures, because the more images are retrieved, the harder to make all these images rank high. Our method can result in better performance in terms of both recall and rank shows that, it not only retrieves the most relevant images, but also all those retrieved images are closer to the top than the other approaches.

In addition, from the aspect of time-cost, our RF mechanism implements faster than SVM, but slower than the query movement approach. Compared with query expansion, our method has the similar time-cost. That is to say, our approach can obtain much better performance without much extra time cost.

5.5 Visualization of Sample Coverage and Feature Re-weighting

Figure 7 shows a 2D visualization of sample coverage and feature re-weighting. For explicitness, only the coverage sets of images in the CRSC are illustrated. Irrelevant images which are used for coverage boundary are also not shown. These figures are obtained from realistic experimental results. The sample coverage are drawn as follows. First, we apply PCA to the image histograms. The projection values on the first two components are taken to get an initial image. Then, we manually move some of the images to make sure that, on the 2D planar, each image in the CRSC can cover all the images in its coverage set.

6 Conclusion

In this paper, we presented a novel RF scheme for CBIR. Our method explicitly aims at selecting a minimum subset of images to cover all the relevant images returned by the system. RankBoost learning is used to maximize each relevant image's coverage set, as well as obtaining a image-specific feature weight distribution. Future research will focus on two directions. One is to build an experiment scenario where user can give detailed relevance score to each relevant image. The other is to explore better image feature representation.

Acknowledgement

This work is supported in part by National Science Foundation of China under grant No. 60673107, National Basic Research Program of China under grant No. 2006CB303100, and a grant from Omron Corporation.

References

1. Rui, Y., Huang, T., Chang, S.: Image retrieval: Current techniques, promising directions, and open issues. J. Vis. Commun. Image R 10(4), 39–62 (1999)
2. Liu, Y., Zhang, D., Lu, G., Ma, W.Y.: A survey of content-based image retrieval with high level semantics. Pattern Recogn. 40, 262–282 (2007)
3. Mezaris, V., Kompatsiaris, I., Strintzis, M.G.: An ontology approach to object-based image retrieval. In: ICIP, vol. 2, pp. 511–514 (2003)
4. Jin, W., Shi, R., Chua, T.S.: A semi-naïve bayesian method incorporating clustering with pair-wise constraints for auto image annotation. In: ACMMM (2004)
5. Tong, S., Chang, E.: Support vector machine active learning for image retrieval. In: ACMMM, pp. 107–118 (2001)
6. Rui, Y., Huang, T., Mehrotra, S.: Content-based image retrieval with relevance feedback in mars. In: ICIP, vol. 2, pp. 815–818 (1997)
7. Ishikawa, Y., Subramanya, R., Faloutsos, C.: Mindreader: Querying databases through multiple examples. In: VLDB, pp. 208–217 (1998)
8. Porkaew, K., Chakrabarti, K.: Query refinement for multimedia similarity retrieval in mars. In: ACMMM, pp. 235–238 (1999)
9. Kim, D.H., Chung, C.W.: Qcluster: Relevance feedback using adaptive clustering for content-based image retrieval. In: SIGMOD, pp. 599–610 (2003)
10. Su, Z., Zhang, H., Li, S., Ma, S.: Relevance feedback in content-based image retrieval: Bayesian framework, feature subspaces, and progressive learning. IEEE T. Image Process 12(8), 924–937 (2003)
11. Sahbi, H., Audibert, J.Y., Keriven, R.: Graph-cut transducers for relevance feedback in content-based image retrieval. In: ICCV (2007)
12. Frome, A., Singer, Y., Sha, F., Malik, J.: Learning globally-consistent local distance functions for shape-based image retrieval and classification. In: ICCV (2007)
13. Bray, N.: Dominating set. In: Weisstein, E.W. (ed.) From MathWorld - A Wolfram Web Resource, `http://mathworld.wolfram.com/DominatingSet.html`
14. Salton, G., Fox, E.A., Wu, H.: Extended boolean information retrieval. Commun. ACM 26(11), 1022–1036 (1983)
15. Racine, K., Yang, Q.: Maintaining unstructured case bases. In: Leake, D.B., Plaza, E. (eds.) ICCBR 1997. LNCS, vol. 1266, pp. 553–564. Springer, Heidelberg (1997)
16. Freund, Y., Iyer, R., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for combining preferences. J. Mach. Learn. Res. 4, 933–969 (2003)
17. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. University of Massachusetts, Amherst, Technical Report 07-49 (October 2007)
18. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from new training examples: An incremental bayesian approach tested on 101 object categories. In: WGMBV (2004)
19. Ahonen, T., Hadid, A., Pietikäinen, M.: Face recognition with local binary patterns. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 469–481. Springer, Heidelberg (2004)

Discriminative Learning for Deformable Shape Segmentation: A Comparative Study

Jingdan Zhang[1,2], Shaohua Kevin Zhou[1], Dorin Comaniciu[1], and Leonard McMillan[2]

[1] Integrated Data Systems Department, Siemens Corporate Research
Princeton, NJ 08540, USA
[2] Department of Computer Science, UNC Chapel Hill
Chapel Hill, NC 27599, USA
{jingdan.zhang,shaohua.zhou,dorin.comaniciu}@siemens.com,
mcmillan@cs.unc.edu

Abstract. We present a comparative study on how to use discriminative learning methods such as classification, regression, and ranking to address deformable shape segmentation. Traditional generative models and energy minimization methods suffer from local minima. By casting the segmentation into a discriminative framework, the target fitting function can be steered to possess a desired shape for ease of optimization yet better characterize the relationship between shape and appearance. To address the high-dimensional learning challenge present in the learning framework, we use a multi-level approach to learning discriminative models. Our experimental results on left ventricle segmentation from ultrasound images and facial feature point localization demonstrate that the discriminative models outperform generative models and energy minimization methods by a large margin.

1 Introduction

Deformable shape segmentation is a long-standing challenge in computer vision and medical imaging. The challenge arises from mainly two aspects: (i) modeling deformable shape and (ii) characterizing the relationship between shape and appearance. Segmentation algorithms must address both aspects successfully. In the paper, we study the latter from a discriminative learning perspective.

Deformable shape can be represented either explicitly or implicitly. Explicit shape representation includes parametric curve/mesh [1], landmark-based model [2], etc. Implicit representation includes level set [3], M-rep [4], etc. In the paper, we focus on landmark-based explicit representation, in which prior knowledge about the shape can be encoded by principal component analysis (PCA), similar to active shape model (ASM) [2]. To have a robust segmentation performance, prior knowledge is important to dealing with the complex shape variations by constraining the shape deformation to a compact model space spanned by a few parameters (e.g., the dominant principal components.). For other shape representations, prior knowledge can be encoded differently, e.g., using bending energy, minimum curve length, etc.

Shape segmentation can be considered as seeking in the model space the shape model best fitting an image. In order to determine how well a hypothesis model matches the

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 711–724, 2008.

image, we need to build a fitting function to characterize the relationship between shape and image appearance. A good fitting function should satisfy two requirements. First, the function can well differentiate the correct solution from its background in the model space. Second, the function can effectively guide the search algorithms to the correct solution.

The fitting function can be learned from a given set of example images with ground truth annotations. In previous work, generative models are commonly used in learning; examples of generative models include ASM [2] and active appearance model (AAM) [5]. Generative models learn the relationship between the ground truth shapes and their appearances to characterize the underlying generating process of data population. However, they have difficulty in satisfying the two requirements mentioned above. A generative function does not typically represent the background and therefore is sub-optimal in differentiating the ground truth shapes from their background. Also generative learning has difficulty in controlling the overall shape of the fitting function. The learned functions often have local extremes, cause difficulties for optimization algorithms.

The fitting function can be also constructed as the energy function in an energy minimization approach, such as active contour model [1] and level set algorithm [6]. The local shape priors, including the elasticity and stiffness of the shape, are crafted into energy functions, which unfortunately also suffer from the same problems as before. They cannot guarantee to produce the lowest energy at the ground truth position and they are likely to have local minima around strong image edges.

Given sufficient training examples, discriminative learning approaches can provide better fitting functions. Discriminate learning has been applied successfully to object detection applications [7,8,9], in which the problem is formulated as a classification problem. In training, the image patches containing the target object are considered as positives, and the patches containing background as negatives. In testing, the target object is detected by scanning, using the trained classifier, the test image over an exhaustive range of similarity transformation. The computational load of the classification approach is proportional to the dimensionality of parameter space. The main challenge of extending the classification approach to deformable shape segmentation is the high dimensionality of the model space, which makes the exhaustive search prohibitive. In [10], a fitting function is learned using a classification method and the learned function is optimized via gradient ascent. The learned fitting function is not smooth and may have local maxima, making it difficult for gradient ascent to find the correct solution.

Recently, discriminative learning has been incorporated into generative models or energy functions to improve the segmentation performance. In [11], boundary detectors are trained to replace the generative models in ASM for better locating the boundary of heart chambers. In [12], a foreground/background classifier is plugged into an energy function to provide the evidence of whether the current pixel belongs to the target object or not. The fitting functions built by these approaches improve the segmentation results. However, they could still have local extremes.

A regression based approach was proposed to learn the fitting function [13]. The target fitting function is constrained to be unimodal and smooth in the model space, which can be used by local optimization algorithms to efficiently estimate the correct solution. The algorithm demonstrated superior performance on segmenting corpus

callosum border from clean/noisy images, segmenting the left ventricle endocardial wall from echocardiogram, and localizing facial feature points.

In this paper, we present a comparative study on how to apply three discriminative learning approaches – classification, regression, and ranking – to deformable shape segmentation. By using discriminative learning in the model space, the fitting function can be learned in a steerable manner. We discuss how to extend the classification approach from object detection to deformable object segmentation. We also propose a ranking based approach to learning the fitting function: The fitting function is trained to produce the highest score around the ground truth and also possesses a desired shape to guide optimization algorithms to the ground truth. To address the high-dimensional learning challenge present in the learning framework, we apply a multi-level approach to learn discriminative models. In section 2, we discuss how to solve the segmentation via classification, regression, and ranking approaches. We also compare these three approaches in terms of learning complexity and computational cost at the segmentation stage. In section 3, we address the common challenge of the three approaches, namely learning in a high dimensional model space. In section 4, we detail the ranking based approach. In section 5, we compare the performance the three approaches on various test datasets.

2 Discriminative Learning Approaches

A shape in an image I can be parameterized by a set of continuous model parameters, C, which contains both rigid and non-rigid components. Given an image I and a hypothesis model C, we can extract a feature image $x(I, C)$ to describe the image appearance associated with C. For conciseness, we use x instead of $x(I, C)$ when there is no confusion in the given context.

There are a variety of ways of building shape models and computing feature images. Though the discriminative learning approaches presented are not bound to a specific shape model, in our implementation we represent a shape by a set of control points. A shape model is built by aligning the training shapes using the generalized Procrustes analysis [2] and applying PCA to the aligned shapes. The model C is defined as $(t_x, t_y, \theta, s, b_1, b_2, \ldots)$, including pose parameters (2D translation, rotation, and scale) and shape parameters corresponding to a reduced set of eigenvectors associated with the largest eigenvalues. We follow the strategy proposed in [13] to obtain a feature image $x(I, C)$ for computational efficiency. For the shape C with M control points, the feature image x is composed by $M + 1$ image patches cropped from I, as shown in Fig. 1. Other approaches involving more computations are can also be used, such as image warping based on linear interpolation [5] or thin plate spline [14].

A supervised learning approach attempts to train a fitting function $f(x(I, C))$ based on a set of training images $\{I\}$ and their corresponding ground truth shape models $\{\underline{C}\}$. The desired output of f is specific to the discriminative approach.

Classification

A classification approach is to learn a classifier f to indicate whether a hypothesis shape C correctly represents the one in an image I or not. The desired output y of f is a binary value. Whether a feature image $x(I, C)$ is positive or negative is determined

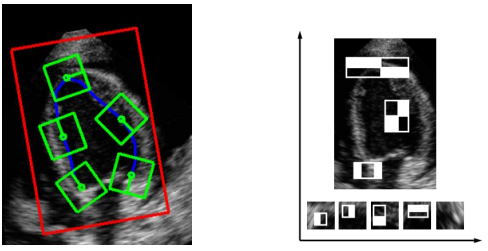


Fig. 1. The feature image x associated with a hypothesis model C. The contour represented by the model C is plotted as blue line. The subimage enclosed by the red box contains global fitness information. The subimges enclosed by the green boxes contain the local fitness information. The image x is composed by subimages with normalized orientation as shown on the right. Examples of Haar-like features are also shown in x.

by the distance between C and the ground truth model $\underline{C}$ in the image I:

$$y = \begin{cases} 1 & \text{if } \| C - \underline{C} \| \leq \epsilon \\ -1, & \text{otherwise} \end{cases}, \tag{1}$$

where ϵ is a threshold that determines the aperture of f. The learned $f(x(I, C))$ is like a boxcar function around the ground truth. Fig. 2(A) shows an ideal learned function f when C is one dimensional. Because the learned f only provides binary indication, the exhaustive search is necessary to estimate the solution, which is computationally prohibitive when the dimensionality of the model C is high.

Regression

A regression approach is to learn a regressor f with real-valued output to indicate the fitness of a hypothesis model C to an image I. The desired output y of f can be designed to facilitate the searching process at the testing stage. In [13], y is set to be a normal distribution:

$$y = \mathcal{N}(C; \underline{C}, \Sigma), \tag{2}$$

where Σ is a covariance matrix determining the aperture of f. The ideally learned f has a smooth and unimodal shape, e.g., a 1D example shown in Fig. 2(B). The function f learned in this way can be effectively optimized by general-purpose local optimization techniques, such as gradient descent or simplex, due to the guidance provided by the gradient of f. However, compared with a classification approach, the desired output is more complicated and, hence, more information needs to be learned at the training stage as it requires the regressor to produce a desired real value for each point in the model space. Learning a regressor in a high-dimensional model space is challenging. Recently, an image-based regression algorithm using boosting methods was proposed in [15] and successfully applied to different applications [16,13].

Ranking

Discriminative learning via ranking is originally proposed to retrieve information based on user preference [17]. In segmentation applications, ranking approaches are used to retrieve candidate shapes from the shape database containing example shapes [18,14].

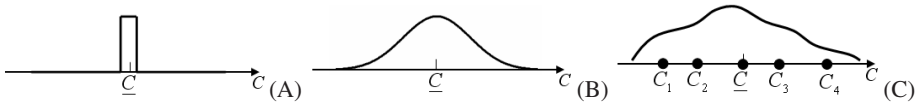


Fig. 2. The learned $f(I, C)$ when C is one dimensional: (A) a classification approach, (B) the regression approach in [13], and (C) the ranking approach. The ground truth of the model is $\underline{C}$.

In this paper, we propose a ranking approach to learning partial ordering of points in the model space. The ordering learned by the ranking function provides essential information to guide the optimization algorithm at the testing stage. Unlike a regression approach, which enforces the regressor to produce an exact value at each point in the model space, ranking only tries to learn relative relations of paired points in the model space. Let (C_0, C_1) be a pair of points in the model space and its associated feature image pair (x_0, x_1). The ordering of x_0 and x_1 is determined by their shape distance to the ground truth: the one closer to the ground truth has a higher order. We learn a ranking function f to satisfy the constraint:

$$\begin{cases} f(x_0) > f(x_1) & \text{if } \| C_0 - \underline{C} \| < \| C_1 - \underline{C} \| \\ f(x_0) < f(x_1) & \text{if } \| C_0 - \underline{C} \| > \| C_1 - \underline{C} \| \\ f(x_0) = f(x_1), & \text{otherwise} \end{cases} \tag{3}$$

Fig. 2(C) illustrates the basic idea of the ranking approach. There are five points in the 1D model space and $\underline{C}$ is the ground truth. At the training stage, a ranking function f is learned to satisfy the ordering constrains: $f(x(I, \underline{C})) > f(x(I, C_2))$, $f(x(I, C_2)) > f(x(I, C_1))$, $f(x(I, \underline{C})) > f(x(I, C_3))$, and $f(x(I, C_3)) > f(x(I, C_4))$. Similar to the regression approach [13], the learned ranking function f is unimodal, which is desired for local optimization techniques. However, the amount of information to be learned in ranking is less than the one in regression. The regression approach learns the full ordering of points in the model space, while the ranking approach only learns partial, pairwise ordering.

We employ the boosting principle to learn our ranking function by selecting relative features to form an additive committee of weak learners. Each weak learner, based on a Haar-like feature that can be computed rapidly, provides a rough ranking. The learned ranking function combines the rough ranking from weak learners and provides the robust ordering information in the shape model space. We will discuss the detailed implementation of the ranking algorithm in section 4.

3 Learning in a High Dimensional Space

The first step toward learning a discriminative function is to sample training examples in the model space. Due to the curse of dimensionality, the number of training examples should be exponential to the model dimensionality to ensure training quality. This poses a big challenge to apply discriminative learning for deformable segmentation applications, in which the dimensionality of the model space is usually high. Another challenge is that it is increasingly difficult to discriminate the correct solution from its

background, when the background points get closer to the solution. In this situation, the image appearance of the background points becomes more and more similar to that of the correct solution. Due to these two challenges, learning a single function in the whole model space to accurately distinguish the solution from its background is ineffective.

We use the multi-level approach proposed in [13] to learn a series of discriminative functions f_k, $k = 1, \ldots, K$, each of which focusing on a region that gradually narrows down to the ground truth. Let Ω_k be the focus region of f_k in the model space, which is defined within an ellipsoid centered at the ground truth:

$$\Omega_k = \{C = (c_1, c_2, ..., c_Q) | \sum_{q=1}^{Q} (c_q - \underline{c}_q)^2 / r_{k,q}^2 \leq 1\} \tag{4}$$

where Q is the dimensionality of the model space and $R_k = (r_{k,1}, \ldots, r_{k,Q})$ defines the range of the focus region. The focus regions are designed to have a nested structure gradually shrinking to the ground truth:

$$\Omega_1 \supset \Omega_2 \supset \ldots \supset \Omega_K \supset \underline{\Omega} \ni \underline{C}, \tag{5}$$

where Ω_1 defines the initial region of the model parameters. It should be big enough to include all the possible solutions in the model space. The final region $\underline{\Omega}$ defines the desired segmentation accuracy.

In segmentation applications, the initial focus region Ω_1 is highly elongated due to the variation in parameter range. It is desirable to first decrease the range of the parameters with a large initial range. The evolution of the range is designed as:

$$r_{k+1,q} = \begin{cases} r_k^{\max}/\gamma & \text{if } r_{k,q} > r_k^{\max}/\gamma \\ r_{k,q} & \text{otherwise} \end{cases}, \tag{6}$$

where $r_k^{\max}$ is the largest value in R_k and γ is a constant controlling the shrinking rate of focus regions (we empirically set $\gamma = 2.9$ for all experiments). Geometrically, the region gradually shrinks from a high-dimensional ellipsoid to a sphere, and then shrinks uniformly thereafter. Fig. 3(A) shows the evolution of the focus regions in a 2D example.

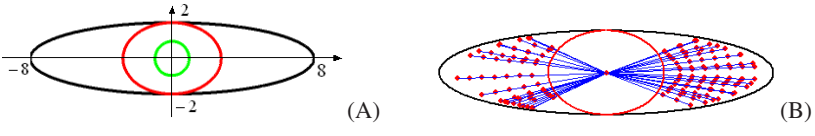


Fig. 3. (A) The three nested focus regions defined by R_1(black), R_2(red) and R_3(green), (B) The result of 2D sampling used for the ranking approach. The lines connect the points on the same ray.

At the testing stage, we apply optimization algorithms sequentially to the learned functions to refine the segmentation results. At the kth stage, we want the solution fallen within the region Ω_k to be pushed into the region Ω_{k+1}. In order to achieve this, the learned function f_k should be able to differentiate the instances in the region Ω_{k+1}

from those in the region $\Omega_k - \Omega_{k+1}$ and provide effective guidance to the optimization algorithms especially in the region $\Omega_k - \Omega_{k+1}$. Data sampling strategies should be accordingly designed.

For the classification approach, the positive examples are sampled from the region Ω_{k+1} and the negatives from the region $\Omega_k - \Omega_{k+1}$. The choice of shrinking rate γ is balanced by two factors. A large γ means a small positive region which requires a fine search grid to detect the solution at the testing stage. This causes high computational expense at the testing stage. On the other hand, a small γ means a large positive region in which the image appearance of the instances has large variation. This causes confusion to the classifier at the training stage.

For the regression approach, gradient sampling is proposed [13]: the learned regressors provide guidance to optimization algorithms based on local gradient. Because the regressor f_k has large gradient in the region $\Omega_k - \Omega_{k+1}$, more training examples are drawn from the region $\Omega_k - \Omega_{k+1}$ to insure the training quality in this region.

The ranking approach is to learn the partial ordering of instances in the model space. Because we perform a line-searching type of optimization, the ordering of instances along the rays starting from the ground truth is the most important. This ordering provides the essential information to guide the optimization algorithms to the ground truth. Also by learning the ordering information from enough rays, the learned ranking function is unimodal which has a global optimum at the ground truth.

We propose a sampling algorithm to sample training pairs for training the ranking function f_k. First, we select a ray starting from the ground truth with random direction. Second, we sample $J + 1$ points $\{C_0, C_1, \ldots, C_J)\}$ on the selected ray, where C_0 is at the ground truth and the remaining J points are sampled from the line segment in the region $\Omega_k - \Omega_{k+1}$. These points are ordered based on the distance to the ground truth. The parameter J is proportional to the length of the line segment. The reason of sampling only from the line segment is that the ordering on this part of the ray is most important for training f_k, which is used to push the solution from the region $\Omega_k - \Omega_{k+1}$ to Ω_{k+1}. Finally, from the training image I, we draw J pairs of training examples $\{(x(I, C_j), x(I, C_{j-1})), j = 1, \ldots, J)\}$, where $x(I, C_{j-1})$ should be ranked above $x(I, C_j)$. We continue this process to sample as much rays as possible that can be fitted into computer memory. Fig. 3(B) shows the sampling result in a 2D model space.

4 Ranking Using Boosting Algorithm

In this section, we present a ranking algorithm based on RankBoost [17] to learn the ordering of the sampled image pairs. Mathematically, we learn a ranking function that minimizes the number of image pairs that are mis-ordered by the learned function. Let Φ be the sampled training set and $(x_0, x_1) \in \Phi$ be a pair of images. Following the sampling strategy proposed in the previous section, x_1 should be ranked above x_0, otherwise a penalty $D(x_0, x_1)$ is imposed. We use equal weighted penalty $D(x_0, x_1)$ in our experiments. The penalty weights can be normalized over the whole training set to a probability distribution $\sum_{(x_0,x_1)\in\Phi} D(x_0, x_1) = 1$.

1. Given: initial distribution D over Φ.
2. Initialize: $D_1 = D$.
3. For $t = 1, 2, \ldots, T$
   - Train weak learner using distribution D_t to get weak ranking g_t.
   - Choose $\alpha_t \in \mathbf{R}$.
   - Update: $D_{t+1}(x_0, x_1) = \frac{D_t(x_0,x_1)\exp[\alpha_t(g_t(x_0)-g_t(x_1))]}{Z_t}$ where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution.)
4. Output the final ranking: $f(x) = \sum_{t=1}^{T} \alpha_t g_t(x)$.

Fig. 4. The RankBoost algorithm [17]

The learning goal is to search for a ranking function f that minimizes the ranking loss,

$$rloss_D(f) = \sum_{(x_0,x_1)\in\Phi} D(x_0, x_1)[\![f(x_0) \geq f(x_1)]\!], \tag{7}$$

where $[\![\pi]\!]$ is defined to be 1 if the predicate π holds and 0 otherwise. In RankBoost, the ranking function $f(x)$ takes an additive form:

$$f_t(x) = f_{t-1}(x) + \alpha_t g_t(x) = \sum_{i=1}^{t} \alpha_i g_i(x), \tag{8}$$

where each $g_i(x)$ is a weak learner residing in a dictionary set $\mathcal{G}$. It maps a feature image x to a real-valued ranking score. The strong learner $f(x)$ combines the weighted ranking scores from weak learners to obtain a robust ranking. Boosting is used to iteratively select weak learners by leveraging the additive nature of $f(x)$: at iteration t, one more additive term $\alpha_t g_t(x)$ is added to the ranking function $f_{t-1}(x)$. The weak learner $g_t(x)$ is selected from the set $\mathcal{G}$ and its associated weight α_t is computed to minimize the ranking loss

$$(g_t, \alpha_t) = \arg \min_{g\in\mathcal{G},\alpha\in\mathbf{R}} \sum_{(x_0,x_1)\in\Phi} D(x_0, x_1)[\![f_{t-1}(x_0)+\alpha g(x_0) \geq f_{t-1}(x_1)+\alpha g(x_1)]\!]. \tag{9}$$

The RankBoost algorithm is shown in Fig. 4. We now discuss the choice of weak learners below.

Weak Learner

The input to a weak learner is a feature image x. We use Haar-like features as primitives to construct the dictionary set $\mathcal{G}$. Each weak learner $g(x)$ is associated with a Haar-like feature $h(x;\eta)$, where η specifies the attribute of the feature, including feature type and window position/size. We further restrict that the features must be contained within one of the image patches in x. Fig. 1 shows some examples of the Haar-like features. By choosing Haar-like features with different attributes, we obtain the over-complete feature representation of the image x. These features can be computed rapidly using a set of pre-computed integral images with different orientations [7].

For each feature $h(x;\eta)$, we use a 1D binary function $g(x;\eta)$ with L bins to produce a weak ranking:

$$g(x;\eta) = \beta_l; \text{ if } h(x;\eta) \in (u_{l-1}, u_l] \tag{10}$$

where $\{u_l; l = 0, \ldots, L\}$ evenly divide the output range of the feature $h(x; \eta)$ to L bins and $\beta_l \in \{0, 1\}$ is the value of the lth bin.

Based on the discussion in [17], when the output of a weak learner has range $[0, 1]$, the weak learner should be trained to maximize the function:

$$r = \sum_{(x_0,x_1)\in\Phi} D(x_0, x_1)(g(x_1; \eta) - g(x_0; \eta)). \tag{11}$$

Following the definition in (10), the function r can be rewritten as

$$r = \sum_{l=1}^{L} \beta_l e_l, \quad e_l := \sum_{h(x_1)\in(u_{l-1},u_l]} D(x_0, x_1) - \sum_{h(x_0)\in(u_{l-1},u_l]} D(x_0, x_1). \tag{12}$$

It is easy to show that in order to maximize r, the lth bin value should be determined as:

$$\beta_l = \begin{cases} 1 & \text{if } e_l > 0 \\ 0, & \text{otherwise} \end{cases}, \tag{13}$$

At each boosting iteration, we exhaust all weak learners and find the one that produces the largest r value. Then the associated weight α is computed as

$$\alpha = \frac{1}{2} \ln \left(\frac{1 + r_{\max}}{1 - r_{\max}} \right). \tag{14}$$

5 Experiments

We tested the proposed discriminative learning approaches on two problems. The boosting principle is used in all three approaches to select and combine weak learners. For classification, we implemented the cascade of boosted binary classifiers based on Adaboost [7], which has been successfully applied to fast object detection. For regression, the regression based on boosting proposed in [13] was applied. To fairly compare these three algorithms, we used the same set of Haar-like features discussed in Section 4. We also compared the three algorithms with other alternative approaches, such as ASM [2] and AAM [5]. In order to enhance the performance of ASM, we also implemented an enhanced ASM version that replaces the regular edge computation by boundary classifiers, which is similar to the approach proposed in [11,12].

To handle the challenge of learning in a high dimensional space, we used the multi-level approach to learn a series of discriminative functions. In training, the initial error ranges of the parameters are set to control the sampling range. The initial error ranges of the shape parameters are assumed to be $3\sqrt{\lambda}$, where λ's are eigenvalues from PCA. We sampled as many training examples as computer memory allows. For our computer with 2GB memory, about 400K training examples are used. In testing, for each test image we randomly generated an initial contour, whose pose parameters are within the error range defined in training and shape parameters are zeros (mean shape). Starting from an initial solution, the learned functions are sequentially applied to refine the solution. For the classification approach, we exhaustively searched around the initial solution and found the candidate having the highest classification probability as the starting point for the next level. For the regression and ranking approaches, we used the simplex optimization method [19] due to its tolerance to shallow maxima caused by image noise.

Table 1. The mean and standard deviation of the segmentation errors. In each cell, there are two rows: the first row reports the mean and standard deviation obtained using all testing data and the second row using 95% of testing data (excluding 5% outliers). For ASM and AAM, we applied multi-resolution searching but only reported the benchmarks of the final results.

(A) LV	ASM	enhanced ASM	Classification	Regression	Ranking
level 1	n/a	n/a	15.85±5.51	11.09±4.31	10.12±3.26
			15.15±4.65	10.43±3.11	9.69±2.69
final level	26.20±17.64	17.91±6.80	14.77±6.53	10.07±4.52	9.93±3.56
	23.43±12.03	17.07±5.82	13.86±5.25	9.41±3.06	9.37±2.62
time (s)	0.94	1.43	18.7	3.15	2.86
(B) AR	AAM		Classification	Regression	Ranking
level 1	n/a		18.28±7.07	17.40±6.34	14.80±5.63
			17.23±5.29	16.59±5.30	13.99±4.49
level 2	n/a		12.94±6.11	8.39±4.09	6.84±2.48
			11.88±3.60	7.72±1.92	6.47±1.87
final level	19.70±23.83		11.66±6.77	5.76±3.99	5.79±2.95
	15.87±17.23		10.53±4.35	5.10±1.38	5.31±2.07
time (s)	0.91		29.4	4.70	3.49

5.1 Endocardial Wall Segmentation: Left Ventricle

Segmenting the endocardial wall in echocardiographic images is a challenging task due to the large shape/appearance variation of the heart chambers and signal dropouts in images. In [13], the fitting functions trained by regression are used to locate the endocardial wall of the left ventricle (LV) in an apical four chamber (A4C) view. In this experiment, we followed the exact setting in [13].

The data set has 528 A4C images from different patients. The LV walls are annotated by experts using contours with 17 control points. The size of the LV in an image is roughly 120×180 pixels. Half of the dataset is used for training and the remaining half for testing. The initial range of the pose is set as $[50, 50, \pi/9, 0.2]$, which means 50 pixels in translation, 20 degrees in rotation, and 20% in scale in the extreme. The model C includes five shape parameters which account for 80% of the total shape variation. In training, two levels of discriminative functions are learned.

The segmentation error is defined as the average Euclidean distance between corresponding control points of the segmented shape and the ground truth. In testing, we set the initial pose of a testing image to be a random perturbation of the ground truth within the initial range $[50, 50, \pi/9, 0.2]$. The average initial error of all testing images is 27.16 pixels. We tested the classification, regression and ranking algorithms, along with ASM and its enhanced version. Table 1(A) shows the mean and standard deviation of the test errors of the above algorithms. The average computational time is also listed in the table. Fig. 5(A) is a plot of the sorted errors, where points on the curve with the same horizontal position do not necessarily correspond to the same test case. Fig. 6 shows some segmentation results using the ranking approach.

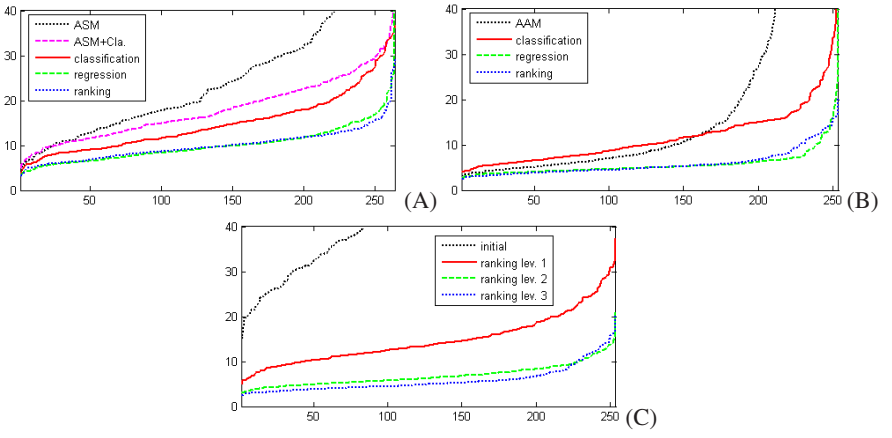


Fig. 5. Sorted errors of the experiment results. The horizontal axes are testing numbers and vertical axes are segmentation errors. (A) LV segmentation, (B) facial feature localization, and (C) the errors of the multi-level refinement using the ranking approach in the LV segmentation.

5.2 Facial Feature Localization

In the second experiment, we tested the performance of the algorithms on the AR face database [20]. There is a total of 508 images with annotations which include 22 control points[1]. The color images were converted to gray-scale. The size of a face is roughly 250×300 in an image. We used half of the data for training and half for testing. Examples of the same subject were not used in both training and testing data. The initial range of the pose is $[100, 100, \pi/9, 0.2]$. The model C includes 5 shape parameters which account for 73% of the total shape variation. In training, we trained three levels of discriminative functions. In testing, the average initial error is 47.19 pixels. We used AAM [21] for comparison. The benchmarks are shown in Table 1(B) and Fig. 5(B). Fig 5(C) shows the errors after each level of refinement when using the ranking functions in testing.

Fig. 7 shows the 2D slices of learned classifiers and ranking functions on a testing image. These slices are obtained by varying the 1st and the 5th parameters of the model in the error range while fixing the remaining parameters as the ground truth, where the 1st is a translation parameter and the 5th is a shape parameter corresponding to the largest eigenvalue. The learned functions have desired shapes which make optimization algorithms perform well on this testing image.

5.3 Discussion

In the two experiments, the segmentation algorithms using discriminative fitting functions consistently outperform the previous algorithm by a large margin. The performance of the ASM algorithm is boosted by using discriminative boundary classifiers;

[1] The annotations are provided by Dr. Cootes, which is available at http://www.isbe.man.ac.uk/~bim.

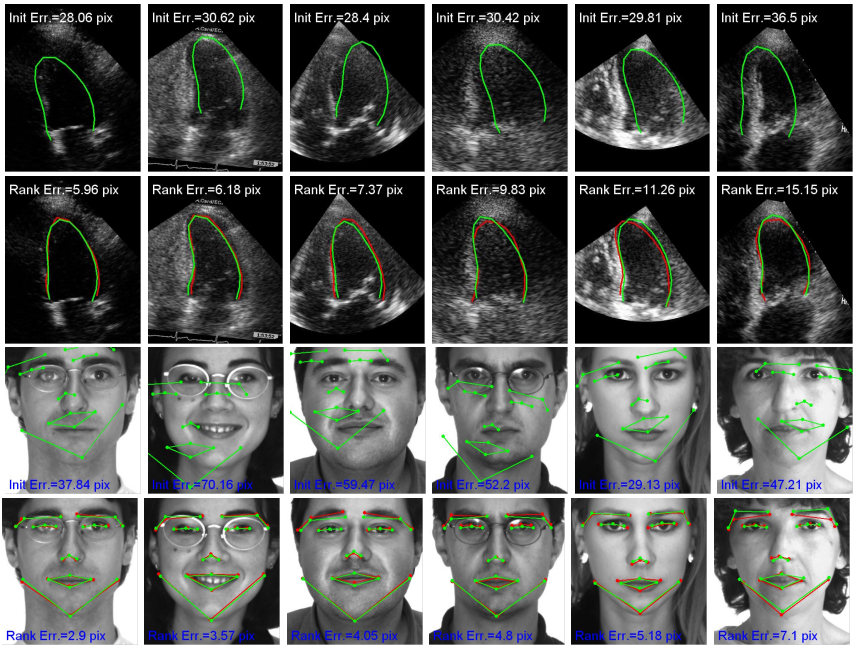


Fig. 6. Segmentation results with a variety of errors obtained by the ranking approach. The first and the third rows show the initial contour position. The second and the fourth rows show the corresponding segmentation results after the multi-level refinement. The initial positions and the results are green line. The ground truths are red line.

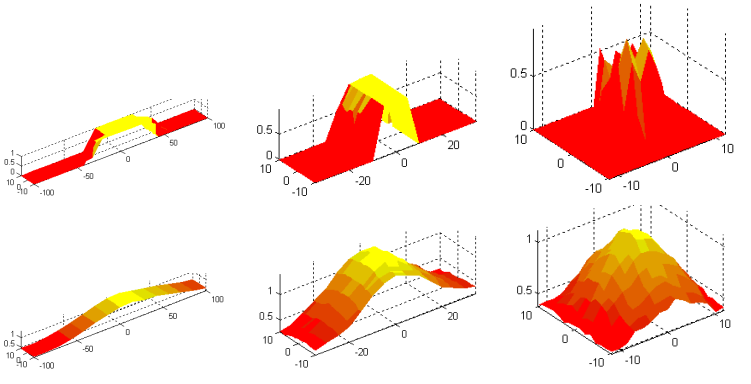


Fig. 7. The 2D slices of the learned classifiers (top row) and ranking functions (bottom row) on a testing image. The left column shows the first level, the middle shows the second level, and the right shows the third level.

however, it still suffers from the local extremes because the boundary classifier is local. The relative poor performance of the classification approach is due to the coarse search grid in exhaustive search. If we use a fine search grid, the segmentation accuracy is

expected to improve. It is interesting to see the performance of the algorithms specifically designed to train classifiers in a high dimensional model space, such as marginal space learning [11]. The ranking approach converges to the correct solution faster than the regression approach does as indicated in the benchmarks of the first and the second level refinement. The main reason might be that ranking only attempts to learn a partial ordering information in the model space and hence its learning complexity is lower than regression. The learned ranking functions are more effective to guide the search algorithm to the correct solution. We will verify this as future work. Like all discriminative learning problems, the discriminative learning approaches suffer from the problem of overfitting especially when the variation of training data cannot totally covers that of testing. Further, the number of sampled data points is hardly sufficient when the model space is high. Because of these problems, the fitting function does not have desired shape on some test data and the local optimization algorithm fails to converge to the ground truth.

Recently, a ranking based algorithm for face alignment was independently proposed [22]. It presents a ranking approach to learning an alignment score function and compares with a classification based algorithm [10]. Compared with the method in [22], we use a different ranking algorithm and apply the multi-level approach to improve the segmentation accuracy. We also compare more algorithms on different kind of data.

6 Conclusions

We have presented a discriminative learning framework for deformable shape segmentation and shown that all three discriminative methods, classification, regression, and ranking, can be applied. We have also addressed how to sample a high-dimensional space and proposed a RankBoost algorithm that does feature selection. Finally, we have demonstrated that the discriminative models outperform generative models and energy minimization methods by a large margin in our experiments on segmentation of left ventricle from ultrasound images and facial feature point localization. In the future, we will further investigate how to sample a high-dimensional space more efficiently and extend this framework to arbitrary shape representations.

References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Int. J. Computer Vision 1(4), 321–331 (1988)
2. Cootes, T., Taylor, C., Cooper, D., Graham, J.: Active shape models–their training and application. Computer Vision and Image Understanding 61(1), 38–59 (1995)
3. Osher, S., Sethian, J.: Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. Journal of Computation Physics 79, 1249 (1988)
4. Pizer, S., Fletcher, P., Joshi, S., Thall, A., Chen, Z., Fridman, Y.: Deformable m-reps for 3D medical image segmentation. Int. J. Computer Vision 55, 85–106 (2003)
5. Cootes, T., Edwards, G., Taylor, C.: Active appearance models. IEEE Trans. Pattern Anal. Machine Intell. 23(6), 681–685 (2001)
6. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Int. J. Computer Vision 22(1), 61–79 (1997)

7. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. CVPR (2001)
8. Georgescu, B., Zhou, X.S., Comaniciu, D., Gupta, A.: Database-guided segmentation of anatomical structures with complex appearance. In: Proc. CVPR (2005)
9. Zhang, J., Zhou, S., McMillan, L., Comaniciu, D.: Joint real-time object detection and pose estimation using probabilistic boosting network. In: Proc. CVPR (2007)
10. Liu, X.: Generic face alignment using boosted appearance model. In: Proc. CVPR (2008)
11. Zheng, Y., Barbu, A., Georgescu, B., Scheuering, M., Comaniciu, D.: Fast automatic heart chamber segmentation from 3D ct data using marginal space learning and steerable features. In: Proc. ICCV (2007)
12. Tu, Z., Zhou, X.S., Comaniciu, D., Luca, B.: A learning based approach for 3D segmentation and colon detagging. In: Proc. European Conf. Computer Vision (2006)
13. Zhang, J., Zhou, S., Comaniciu, D., McMillan, L.: Conditional density learning via regression with application to deformable shape segmentation. In: Proc. CVPR (2008)
14. Zheng, Y., Zhou, X.S., Georgescu, B., Zhou, S., Comaniciu, D.: Example based non-rigid shape detection. In: Proc. European Conf. Computer Vision (2006)
15. Zhou, S., Gerogescu, B., Zhou, X., Comaniciu, D.: Image-based regression using boosting methods. In: Proc. ICCV (2005)
16. Zhou, S., Comaniciu, D.: Shape regression machine. In: Proc. IPMI (2007)
17. Freund, Y., Iyer, R., Schapire, R., Singer, Y.: An efficient boosting algorithm for combining preferences. J. Machine Learning Research 4(6), 933–970 (2004)
18. Athitsos, V., Alon, J., Sclaroff, S., Kollios, G.: Boostmap: A method for efficient approximate similarity rankings. In: Proc. CVPR (2004)
19. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Numerical recipes in C, 2nd edn. Cambridge University Press, Cambridge (1992)
20. Martinez, A.M., Benavente, R.: The AR face database (1998)
21. Stegmann, M.B., Ersboll, B.K., Larsen, R.: FAME–a flexible appearance modeling environment. IEEE Trans. Medical Imaging 22(10), 1319–1331 (2003)
22. Wu, H., Liu, X., Doretto, G.: Face alignment via boosted ranking model. In: Proc. CVPR (2008)

Discriminative Locality Alignment

Tianhao Zhang[1], Dacheng Tao[2,3], and Jie Yang[1]

[1] Institute of Image Processing and Pattern Recognition,
Shanghai Jiao Tong University, Shanghai, China
[2] School of Computer Engineering, Nanyang Technological University,
50 Nanyang Avenue, Singapore
[3] College of Computer Science, Zhejiang University, China
z.tianhao@gmail.com, dacheng.tao@gmail.com, jieyang@sjtu.edu.cn

Abstract. Fisher's linear discriminant analysis (LDA), one of the most popular dimensionality reduction algorithms for classification, has three particular problems: it fails to find the nonlinear structure hidden in the high dimensional data; it assumes all samples contribute equivalently to reduce dimension for classification; and it suffers from the matrix singularity problem. In this paper, we propose a new algorithm, termed Discriminative Locality Alignment (DLA), to deal with these problems. The algorithm operates in the following three stages: first, in part optimization, discriminative information is imposed over patches, each of which is associated with one sample and its neighbors; then, in sample weighting, each part optimization is weighted by the *margin degree*, a measure of the importance of a given sample; and finally, in whole alignment, the alignment trick is used to align all weighted part optimizations to the whole optimization. Furthermore, DLA is extended to the semi-supervised case, i.e., semi-supervised DLA (SDLA), which utilizes unlabeled samples to improve the classification performance. Thorough empirical studies on the face recognition demonstrate the effectiveness of both DLA and SDLA.

1 Introduction

Dimensionality reduction is the process of transforming data from a high dimensional space to a low dimensional space to reveal the intrinsic structure of the distribution of data. It plays a crucial role in the field of computer vision and pattern recognition as a way of dealing with the "curse of dimensionality". In past decades, a large number of dimensionality reduction algorithms have been proposed and studied. Among them, principal components analysis (PCA) [9] and Fisher's linear discriminant analysis (LDA) [6] are two of the most popular linear dimensionality reduction algorithms.

PCA [9] maximizes the mutual information between original high dimensional Gaussian distributed data and projected low dimensional data. PCA is optimal for reconstruction of Gaussian distributed data. However it is not optimal for classification [14] problems. LDA overcomes this shortcoming by utilizing the class label information. It finds the projection directions that maximize the trace

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 725–738, 2008.

of the between-class scatter matrix and minimize the trace of the within-class scatter matrix simultaneously. While LDA is a good algorithm to be applied for classification, it also has several problems as follows.

First, LDA considers only the global Euclidean structure, so it cannot discover the nonlinear structure hidden in the high dimensional non-Gaussian distributed data. Numerous manifold learning algorithms have been developed as a promising tool for analyzing the high dimensional data that lie on or near a submanifold of the observation space. Representative works include locally linear embedding (LLE) [11], Laplacian eigenmaps (LE) [2], local tangent space alignment (LTSA) [19], locality preserving projections (LPP) [8]. These algorithms, which aim to preserve the local geometry of samples, can attack the nonlinear distribution of data.

Second, LDA is in fact based on the assumption that all samples contribute equivalently for discriminative dimensionality reduction, although samples around the margins, i.e., marginal samples, are more important in classification than inner samples. A recently developed algorithm, which breaks through the assumption of equal contributions, is marginal Fisher analysis (MFA) [16]. MFA uses only marginal samples to construct the penalty graph which characterizes the interclass separability. However, it does not give these marginal samples specific weights to describe how important each is. Furthermore, MFA may lose discriminative information since it completely ignores inner samples in constructing the penalty graph.

Finally, LDA suffers from the matrix singularity problem since the between-class scatter matrix is often singular. Many algorithms have been proposed to deal with this, such as PCA plus LDA [1], direct LDA (DLDA) [18], and null-space LDA (NLDA) [5]. However, all of them may fail to consider all possible discriminative information in selecting discriminative subspace.

Most importantly, in the context of this work, almost all existing variants of LDA respond to just one or two of LDA's suite of problems, yet they may remain open to others. In order to overcome all aforementioned problems in LDA simultaneously, a new linear algorithm termed Discriminative Locality Alignment (DLA) is proposed. The algorithm operates in the following three stages: 1) the part optimization stage, 2) the sample weighting stage, and 3) the whole alignment stage. First, discriminative information is imposed over patches, each of which is associated with one sample and its neighbors; then each part optimization is weighted by *margin degree*, a measure of the importance of a given sample for classification; and finally the alignment trick [19,20,21] is used to align all of the weighted part optimizations to the whole optimization.DLA has three particular advantages: 1) because it focuses on the local patch of each sample, it can deal with the nonlinearity of the distribution of samples while preserving the discriminative information; 2) since the importance of marginal samples is enhanced for discriminative subspace selection, it learns low dimensional representations for classification properly; and 3) because it obviates the need to compute the inverse of a matrix, it has no the matrix singularity problem. In addition, we extend DLA

to the semi- supervised learning case, i.e., semi-supervised DLA (SDLA), by incorporating the part optimizations of the unlabeled samples.

The rest of the paper is organized as follows. Section 2 details the proposed DLA algorithm. Section 3 extends DLA to SDLA. Section 4 gives our experimental results. Section 5 concludes.

2 Discriminative Locality Alignment

Consider a set of samples $X = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N] \in \mathbb{R}^{m \times N}$, and each sample $\boldsymbol{x}_i$ belongs to one of the C classes. The problem of linear dimensionality reduction is to find a projection matrix U that maps $X \in \mathbb{R}^{m \times N}$ to $Y \in \mathbb{R}^{d \times N}$, i.e., $Y = U^T X$, where $d < m$. In this section, Discriminative Locality Alignment (DLA) is proposed to overcome problems in LDA for linear dimensionality reduction.

DLA operates in three stages. In the first stage, for each sample in the dataset, one patch is built by the given sample and its neighbors which including the samples from not only a same class but also different classes from the given sample. On each patch, an objective function is designed to preserve the local discriminative information. Since each sample can be seen as a part of the whole dataset, the stage is termed "part optimization". In the second stage, *margin degree* is defined for each sample as a measure of the sample importance in contributing classification. Then, each part optimization obtained from the first stage is weighted based on the *margin degree*. The stage termed "sample weighting". In the final stage, termed "whole alignment", all the weighted part optimizations are integrated into together to form a global coordinate according to the alignment trick [19,20,21]. The projection matrix can be obtained by solving a standard eigen-decomposition problem.

2.1 Part Optimization

For a given sample $\boldsymbol{x}_i$, according to the class label information, we can divide the other ones into the two groups: samples in the same class with $\boldsymbol{x}_i$ and samples from different classes with $\boldsymbol{x}_i$. We can select k_1 nearest neighbors with respect to $\boldsymbol{x}_i$ from samples in the same class with $\boldsymbol{x}_i$ and term them *neighbor samples from an identical class*: $\boldsymbol{x}_{i^1}, \cdots, \boldsymbol{x}_{i^{k_1}}$. We select k_2 nearest neighbors with respect to $\boldsymbol{x}_i$ from samples in different classes with $\boldsymbol{x}_i$ and term them *neighbor samples from different classes*: $\boldsymbol{x}_{i_1}, \cdots, \boldsymbol{x}_{i_{k_2}}$. The local patch for the sample $\boldsymbol{x}_i$ is constructed by putting $\boldsymbol{x}_i$, $\boldsymbol{x}_{i^1}, \cdots, \boldsymbol{x}_{i^{k_1}}$, and $\boldsymbol{x}_{i_1}, \cdots, \boldsymbol{x}_{i_{k_2}}$ together as $X_i = [\boldsymbol{x}_i, \boldsymbol{x}_{i^1}, \cdots, \boldsymbol{x}_{i^{k_1}}, \boldsymbol{x}_{i_1}, \cdots, \boldsymbol{x}_{i_{k_2}}]$.

For each patch, the corresponding output in the low dimensional space is $Y_i = [\boldsymbol{y}_i, \boldsymbol{y}_{i^1}, \cdots, \boldsymbol{y}_{i^{k_1}}, \boldsymbol{y}_{i_1}, \cdots, \boldsymbol{y}_{i_{k_2}}]$. In the low dimensional space, we expect that distances between the given sample and *neighbor samples from an identical class* are as small as possible, while distances between the given sample and *neighbor samples from different classes* are as large as possible, as illustrated in Figure 1. The left part of the figure shows the i^{th} patch in the original high dimensional space and the patch consists of $\boldsymbol{x}_i$, *neighbor samples from an identical class* (i.e.,

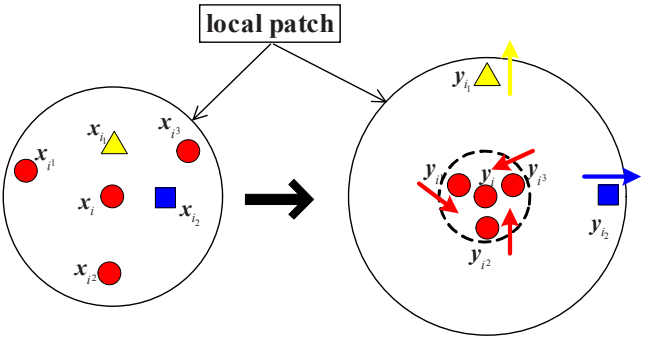


Fig. 1. The part optimization stage in DLA

$\boldsymbol{x}_{i^1}$, $\boldsymbol{x}_{i^2}$, and $\boldsymbol{x}_{i^3}$), and *neighbor samples from different classes* (i.e., $\boldsymbol{x}_{i_1}$ and $\boldsymbol{x}_{i_2}$). The expected results on the patch in the low dimensional space are shown as the right part of the figure. Low dimensional samples $\boldsymbol{y}_{i^1}$, $\boldsymbol{y}_{i^2}$, and $\boldsymbol{y}_{i^3}$ are as close as possible to $\boldsymbol{y}_i$, while low dimensional samples $\boldsymbol{y}_{i_1}$ and $\boldsymbol{y}_{i_2}$ are as far as possible away from $\boldsymbol{y}_i$.

For each patch in the low dimensional space, we expect that distances between $\boldsymbol{y}_i$ and *neighbor samples from an identical class* are as small as possible, so we have:

$$\arg\min_{\boldsymbol{y}_i} \sum_{j=1}^{k_1} \|\boldsymbol{y}_i - \boldsymbol{y}_{i^j}\|^2. \tag{1}$$

Meanwhile, we expect that distances between $\boldsymbol{y}_i$ and *neighbor samples from different classes* are as large as possible, so we have:

$$\arg\max_{\boldsymbol{y}_i} \sum_{p=1}^{k_2} \left\|\boldsymbol{y}_i - \boldsymbol{y}_{i_p}\right\|^2. \tag{2}$$

Since the patch built by the local neighborhood can be regarded approximately Euclidean [11], we formulate the part discriminator by using the linear manipulation:

$$\arg\min_{\boldsymbol{y}_i} \left(\sum_{j=1}^{k_1} \|\boldsymbol{y}_i - \boldsymbol{y}_{i^j}\|^2 - \beta \sum_{p=1}^{k_2} \|\boldsymbol{y}_i - \boldsymbol{y}_{i_p}\|^2 \right), \tag{3}$$

where β is a scaling factor in $[0, 1]$ to unify the different measures of the within-class distance and the between-class distance. Define the coefficients vector

$$\boldsymbol{\omega}_i = \left[\overbrace{1, \cdots, 1,}^{k_1} \quad \overbrace{-\beta, \cdots, -\beta}^{k_2} \right]^T, \tag{4}$$

then, Eq. (3) reduces to:

$$\begin{aligned}
&\arg\min_{\boldsymbol{y}_i}\left(\sum_{j=1}^{k_1}\|\boldsymbol{y}_i-\boldsymbol{y}_{i^j}\|^2(\boldsymbol{\omega}_i)_j+\sum_{p=1}^{k_2}\|\boldsymbol{y}_i-\boldsymbol{y}_{i_p}\|^2(\boldsymbol{\omega}_i)_{p+k_1}\right)\\
=&\arg\min_{\boldsymbol{y}_i}\left(\sum_{j=1}^{k_1+k_2}\left\|\boldsymbol{y}_{F_i\{1\}}-\boldsymbol{y}_{F_i\{j+1\}}\right\|^2(\boldsymbol{\omega}_i)_j\right)\\
=&\arg\min_{Y_i}\operatorname{tr}\left(Y_i\begin{bmatrix}-\boldsymbol{e}_{k_1+k_2}^T\\ I_{k_1+k_2}\end{bmatrix}\operatorname{diag}(\boldsymbol{\omega}_i)\begin{bmatrix}-\boldsymbol{e}_{k_1+k_2}\ I_{k_1+k_2}\end{bmatrix}Y_i^T\right)\\
=&\arg\min_{Y_i}\operatorname{tr}\left(Y_iL_iY_i^T\right),
\end{aligned}\tag{5}$$

where $F_i=\left\{i,i^1,\cdots,i^{k_1},i_1,\cdots,i_{k_2}\right\}$ is the index set for the i^{th} patch; $\boldsymbol{e}_{k_1+k_2}=[1,\cdots,1]^T\in\mathbb{R}^{k_1+k_2}$; $I_{k_1+k_2}$ is the $(k_1+k_2)\times(k_1+k_2)$ identity matrix; $\operatorname{diag}(\cdot)$ is the diagonalization operator; L_i encapsulates both the local geometry and the discriminative information, and it is given by

$$L_i=\begin{bmatrix}\sum_{j=1}^{k_1+k_2}(\boldsymbol{\omega}_i)_j & -\boldsymbol{\omega}_i^T\\ -\boldsymbol{\omega}_i & \operatorname{diag}(\boldsymbol{\omega}_i)\end{bmatrix}.\tag{6}$$

2.2 Sample Weighting

In general, samples around classification margins have a higher risk of being misclassified than samples far away from margins. As shown in Figure 2, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, which are lying around the nearby classification margin, are more important than $\boldsymbol{x}_3$ in seeking a subspace for classification.

To quantify the importance of a sample $\boldsymbol{x}_i$ for discriminative subspace selection, we need to find a measure, termed *margin degree* m_i. For a sample, its *margin degree* should be proportional to the number of samples with different class labels from the label of the sample but in the ϵ-ball centered at the sample. Therefore, a possible definition of the *margin degree* m_i for the i^{th} sample $\boldsymbol{x}_i$ is

$$m_i=\exp\left(-\frac{1}{(n_i+\delta)t}\right)\qquad i=1,\cdots,N,\tag{7}$$

where n_i is the number of samples $\boldsymbol{x}_j$ in the ϵ-ball centered at $\boldsymbol{x}_i$ with labels $l(\boldsymbol{x}_j)$ different from the label of $\boldsymbol{x}_j$; $l(\boldsymbol{x})$ is the class label of the sample $\boldsymbol{x}$; δ is a regularization parameter; and t is a scaling factor. In Figure 2, for a fixed ϵ, $n_1=4$ because there are 4 samples with different class labels from that of $\boldsymbol{x}_1$ in the ϵ-ball centered at $\boldsymbol{x}_1$; $n_2=1$ because there are 1 sample with different class label from that of $\boldsymbol{x}_2$ in the ϵ-ball centered at $\boldsymbol{x}_2$; $n_3=0$ because there are no sample with different class label from that of $\boldsymbol{x}_3$ in the ϵ-ball centered at $\boldsymbol{x}_3$. According to Eq.(7), the corresponding *margin degrees* of these three samples are ordered as $m_1>m_2>m_3$.

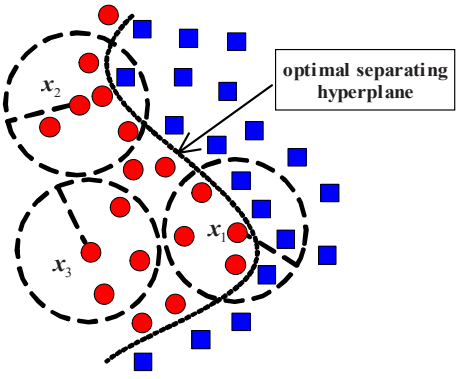


Fig. 2. Illustration for sample weighting

In DLA, the part optimization of the i^{th} patch is weighted by the *margin degree* of the i^{th} sample before the whole alignment stage, i.e.,

$$\arg\min_{Y_i} m_i \mathrm{tr}\left(Y_i L_i Y_i^T\right) = \arg\min_{Y_i} \mathrm{tr}\left(Y_i m_i L_i Y_i^T\right). \tag{8}$$

2.3 Whole Alignment

For each patch X_i, $i = 1, \cdots, N$, we have the weighted part optimizations described as Eq. (8). In this subsection, these optimizations will be unified together as a whole one by assuming that the coordinate for the i^{th} patch $Y_i = \left[\boldsymbol{y}_i, \boldsymbol{y}_{i1}, \cdots, \boldsymbol{y}_{ik_1}, \boldsymbol{y}_{i_1}, \cdots, \boldsymbol{y}_{i_{k_2}}\right]$ is selected from the global coordinate $Y = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_N]$, such that

$$Y_i = Y S_i, \tag{9}$$

where $S_i \in \mathbb{R}^{N \times (k_1+k_2+1)}$ is the selection matrix and an entry is defined as:

$$(S_i)_{pq} = \begin{cases} 1 \text{ if } p = F_i\{q\} \\ 0 \quad \text{else}. \end{cases} \tag{10}$$

Then, Eq. (8) can be rewritten as:

$$\arg\min_{Y} \mathrm{tr}\left(Y S_i m_i L_i S_i^T Y^T\right). \tag{11}$$

By summing over all part optimizations described as Eq. (11) together, we can obtain the whole alignment as:

$$\begin{aligned} \arg\min_{Y} \sum_{i=1}^{N} \mathrm{tr}\left(Y S_i m_i L_i S_i^T Y^T\right) &= \arg\min_{Y} \mathrm{tr}\left(Y \left(\sum_{i=1}^{N} S_i m_i L_i S_i^T\right) Y^T\right) \\ &= \arg\min_{Y} \mathrm{tr}\left(Y L Y^T\right), \end{aligned} \tag{12}$$

where $L = \sum_{i=1}^{N} S_i m_i L_i S_i^T \in \mathbb{R}^{N \times N}$ is the alignment matrix [19]. It is obtained based on an iterative procedure:

$$L\left(F_i, F_i\right) \leftarrow L\left(F_i, F_i\right) + m_i L_i, \tag{13}$$

for $i = 1, \cdots, N$, with the initialization $L = 0$.

To obtain the linear and orthogonal projection matrix U, such as $Y = U^T X$, we can impose $U^T U = I_d$, where I_d is the $d \times d$ identity matrix. Eq. (12) is deformed as:

$$\arg\min_{U} \operatorname{tr}\left(U^T X L X^T U\right) \text{ s.t. } U^T U = I_d. \tag{14}$$

Obviously, solutions of Eq.(14) are given by using the standard eigen-decomposition:

$$X L X^T \boldsymbol{u} = \lambda \boldsymbol{u}. \tag{15}$$

Let the column vectors $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_d$ be the solutions of Eq. (15), ordered according to eigenvalues, $\lambda_1 < \lambda_2 < \cdots < \lambda_d$. The optimal projection matrix U is then given by: $U = [\boldsymbol{u}_1, \boldsymbol{u}_2 \cdots, \boldsymbol{u}_d]$.

Different from algorithms, e.g., LDA [1], LPP [8], and MFA [16], which lead to a generalized eigenvalue problem, DLA successfully avoids the matrix singularity problem since it has no inverse operation over a matrix. However, the PCA step is still recommended to reduce noise. The procedure of DLA is listed as following:

1. Use PCA to project the dataset X into the subspace for eliminating the useless information. To make it clear, we still use X to denote the dataset in the PCA subspace in the following steps. We denote by U_{PCA} the PCA projection matrix;
2. For each sample $\boldsymbol{x}_i$ in dataset X, $i = 1, \cdots, N$, search k_1 *neighbor samples from an identical class* and k_2 neighbor samples from different classes, and then build the patch $X_i = [\boldsymbol{x}_i, \boldsymbol{x}_{i1}, \cdots, \boldsymbol{x}_{i^{k_1}}, \boldsymbol{x}_{i_1}, \cdots, \boldsymbol{x}_{i_{k_2}}]$;
3. Compute L_i by Eq. (6), and m_i by Eq. (7). Construct the alignment matrix L by the iterative procedure described by Eq. (13); and
4. Solve the standard eigen-decomposition: $X L X^T \boldsymbol{u} = \lambda \boldsymbol{u}$ to obtain the DLA projection matrix $U_{DLA} = [\boldsymbol{u}_1, \boldsymbol{u}_2 \cdots, \boldsymbol{u}_d]$, whose vectors are the eigenvectors corresponding to the d smallest eigenvalues. The final projection matrix is as follows: $U = U_{PCA} U_{DLA}$.

3 Semi-supervised DLA

Recent researches [3,22] show that unlabeled samples may be helpful to improve the classification performance. In this section, we generalize DLA by introducing new part optimizations by taking unlabeled samples into account and then incorporating them to the whole alignment stage as semi-supervised DLA (SDLA). The unlabeled samples are attached to the original labeled samples as: $X = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N, \boldsymbol{x}_{N+1}, \cdots, \boldsymbol{x}_{N+N_U}]$, where the first N samples are labeled and the left N_U ones are unlabeled. The part optimization for each labeled sample is given by Eq. (8).

Unlabeled samples are valuable to enhance the local geometry. For each unlabeled sample $\boldsymbol{x}_i,\ \ i = N+1, \cdots, N+N_U$, we search its k_S nearest neighbors $\boldsymbol{x}_{i_1}, \cdots, \boldsymbol{x}_{i_{k_S}}$ in all training samples including both labeled and unlabeled ones. Let $X_i = [\boldsymbol{x}_i, \boldsymbol{x}_{i_1}, \cdots, \boldsymbol{x}_{i_{k_S}}]$ denote the i^{th} patch and the associated index set is given by $F_i^U = \{i, i_1, \cdots, i_{k_S}\}$. To capture the local geometry of the i^{th} patch, we expect nearby samples remain nearby, or $\boldsymbol{y}_i \in \mathbb{R}^d$ is close to $\boldsymbol{y}_{i_1}, \cdots, \boldsymbol{y}_{i_{k_S}}$, i.e.,

$$\begin{aligned}
&\arg\min_{\boldsymbol{y}_i} \sum_{j=1}^{k_S} \left\| \boldsymbol{y}_i - \boldsymbol{y}_{i_j} \right\|^2 \\
=\ &\arg\min_{\boldsymbol{y}_i} \operatorname{tr} \left(\begin{bmatrix} (\boldsymbol{y}_i - \boldsymbol{y}_{i_1})^T \\ \vdots \\ \left(\boldsymbol{y}_i - \boldsymbol{y}_{i_{k_S}}\right)^T \end{bmatrix} \left[\boldsymbol{y}_i - \boldsymbol{y}_{i_1}, \cdots, \boldsymbol{y}_i - \boldsymbol{y}_{i_{k_S}} \right] \right) \\
=\ &\arg\min_{Y_i} \operatorname{tr} \left(Y_i \begin{bmatrix} -\boldsymbol{e}_{k_S}^T \\ I_{k_S} \end{bmatrix} \left[-\boldsymbol{e}_{k_S}\ I_{k_S} \right] Y_i^T \right) \\
=\ &\arg\min_{Y_i} \operatorname{tr} \left(Y_i L_i^U Y_i^T \right),
\end{aligned} \tag{16}$$

where, $\boldsymbol{e}_{k_S} = [1, \cdots, 1]^T \in \mathbb{R}^{k_S}$; I_{k_S} is the $k_S \times k_S$ identity matrix; and

$$L_i^U = \begin{bmatrix} -\boldsymbol{e}_{k_S}^T \\ I_{k_S} \end{bmatrix} \left[-\boldsymbol{e}_{k_S}\ I_{k_S} \right] = \begin{bmatrix} k_s & -\boldsymbol{e}_{k_S}^T \\ -\boldsymbol{e}_{k_S} & I_{k_S} \end{bmatrix}. \tag{17}$$

Since the unlabeled samples cannot provide the margin information, the sample weighting stage is omitted for unlabeled ones in SDLA. Putting all samples together, we have:

$$\begin{aligned}
&\arg \sum_{i=1}^{N} \min_{Y_i} \operatorname{tr} \left(Y_i m_i L_i Y_i^T \right) + \gamma \arg \sum_{i=N+1}^{N+N_U} \min_{Y_i} \operatorname{tr} \left(Y_i L_i^U Y_i^T \right) \\
=\ &\arg\min_{Y} \operatorname{tr} \left(Y \left(\sum_{i=1}^{N} S_i^L m_i L_i \left(S_i^L\right)^T + \sum_{i=N+1}^{N+N_U} S_i^U \gamma L_i^U \left(S_i^U\right)^T \right) Y^T \right) \\
=\ &\arg\min_{Y} \operatorname{tr} \left(Y L^S Y^T \right),
\end{aligned} \tag{18}$$

where γ is a control parameter; $S_i^L \in \mathbb{R}^{(N+N_U)\times(k_1+k_2+1)}$and $S_i^U \in \mathbb{R}^{(N+N_U)\times(k_S+1)}$ are the selection matrices defined similarly as in Section 2.3; and $L^S \in \mathbb{R}^{(N+N_U)\times(N+N_U)}$ is the alignment matrix constructed by

$$\begin{cases} L^S\left(F_i, F_i\right) \leftarrow L^S\left(F_i, F_i\right) + m_i L_i, \text{ for } i = 1, \cdots, N \\ L^S\left(F_i^U, F_i^U\right) \leftarrow L^S\left(F_i^U, F_i^U\right) + \gamma L_i^U, \text{ for } i = N+1, \cdots, N+N_U, \end{cases} \tag{19}$$

with the initialization $L^S = 0$.

Similar to Section 2.3, the problem is converted to a standard eigenvalue decomposition: $XL^SX^T\boldsymbol{u} = \lambda\boldsymbol{u}$. The projection matrix U_{SDLA} contains eigenvectors associated with the d smallest eigenvalues. Similar to DLA, PCA is also utilized to reduce sample noise, and the final projection matrix is $U = U_{PCA}U_{SDLA}$.

4 Experiments

In this section, we compare the proposed DLA algorithm against representative dimensionality reduction algorithms, e.g., PCA [15], LDA [1], SLPP (LPP1 in [4]), and MFA [16]. We also study the performance of DLA by varying parameters k_1 (the number of *neighbor samples from an identical class*) and k_2 (the number of *neighbor samples from different classes*) which are crucial in building patches. Finally, the SDLA algorithm is evaluated by comparing with the original DLA. To begin with, we briefly introduce the three steps for the recognition problems.

First, we perform each of the involved algorithms on training samples to learn projection matrices. Second, each testing sample is projected to a low dimensional subspace via a projection matrix. Finally, the *nearest neighbor* (NN) classifier is used to recognize testing samples in the projected subspace.

4.1 Data

Three face image databases: UMIST [7], YALE [1], and FERET [10] are utilized for empirical study. The UMIST database consists of 564 face images from 20 subjects. The individuals are a mix of race, sex and appearance and are photographed in a range of poses from profile to frontal views. The YALE database contains face images collected from 15 individuals, 11 images for each individual and showing varying facial expressions and configurations. The FERET database contains 13,539 face images from 1,565 subjects. The images vary in size, pose, illumination, facial expression and age.

For UMIST and YALE, all face images are used in the experiments. For FERET, we randomly select 100 individuals, each of which has 7 images. All images from three databases are cropped with reference to the eyes and cropped images are normalized to 40×40 pixel arrays with 256 gray levels per pixel.

Fig. 3. Sample images. The first row comes from UMIST [7]; the second row comes from YALE [1]; and the third row comes from FERET [10].

Figure 3 shows sample images from these three databases. Each image is reshaped to one long vector by arranging its pixel values in a fixed order.

4.2 General Experiments

We compare the proposed DLA with two different settings, i.e., DLA1 and DLA2, to well-known related dimensionality reduction algorithms, which are PCA [15], LDA [1], SLPP (LPP1 in [4]), and MFA [16], in terms of effectiveness. For DLA1, we set $t = \infty$ in Eq. (7), while in DLA2, t is determined empirically.

For all algorithms except PCA, the first step is PCA projection. In the following experiments, we project samples to the PCA subspace with $N-1$ dimensions for SLPP [4], DLA1, and DLA2. For LDA [1] and MFA [16], we retain $N - C$ dimensions in the PCA step.

For UMIST and YALE, we randomly select p (= 3, 5, 7) images per individual for training, and use the remaining images for testing. For FERET, p (= 3, 4, 5) images per individual are selected for training, and the remaining for testing. All trials are repeated ten times, and then the average recognition results are calculated. Figure 4 shows plots of recognition rate versus dimensionality reduction on three databases. Table 1 lists the best recognition rate for each algorithm. It also provides the optimal values of k_1 and k_2 for DLA1 and DLA2, which crucial since they have the special sense for building patches.

It is shown that both DLA1 and DLA2 outperform conventional algorithms. DLA2 performs better than DLA1 because weights over part optimizations based *margin degree* are considered to benefit the discriminative subspace selection.

It is worth emphasizing that LDA, SLPP and MFA perform poorly on FERET because face images from FERET are more complex and contain more interference for identification. One method enhance their performance is removing such useless information by using PCA projection retaining appropriate percent energies. We also conduct experiments on FERET by exploring all possible PCA

Table 1. Best recognition rates (%) on three databases. For PCA, LDA SLPP, and MFA, the numbers in the parentheses are the subspace dimensions. For DLA1 and DLA2, the first numbers in the parentheses are the subspace dimensions, the second and the third numbers are k_1 and k_2, respectively. Numbers in the second column denote the number of training samples per subject.

		PCA	LDA	SLPP	MFA	DLA1	DLA2
UMIST	3	71.62(59)	79.71(18)	76.58(19)	82.64(11)	84.89(18,2,1)	**86.78**(18,2,1)
	5	82.88(99)	88.51(19)	86.06(19)	92.61(14)	93.85(10,3,4)	**95.20**(10,3,4)
	7	90.53(135)	93.31(19)	91.36(19)	94.28(19)	97.01(33,4,5)	**97.45**(33,4,5)
YALE	3	52.33(44)	64.08(14)	67.00(13)	64.33(12)	68.50(18,2,1)	**69.67**(18,2,1)
	5	58.33(74)	72.78(14)	73.44(14)	73.44(15)	78.11(30,3,4)	**79.89**(30,3,4)
	7	63.33(36)	80.80(13)	82.33(14)	82.67(15)	83.83(15,3,5)	**86.50**(15,3,5)
FERET	3	41.41(107)	51.18(38)	49.55(99)	55.32(47)	84.62(24,1,3)	**86.32**(24,1,3)
	4	47.00(102)	53.40(42)	53.66(99)	58.27(41)	91.87(25,3,5)	**93.03**(25,3,5)
	5	51.55(87)	53.60(50)	54.75(96)	58.65(62)	92.85(23,2,5)	**94.33**(23,2,5)

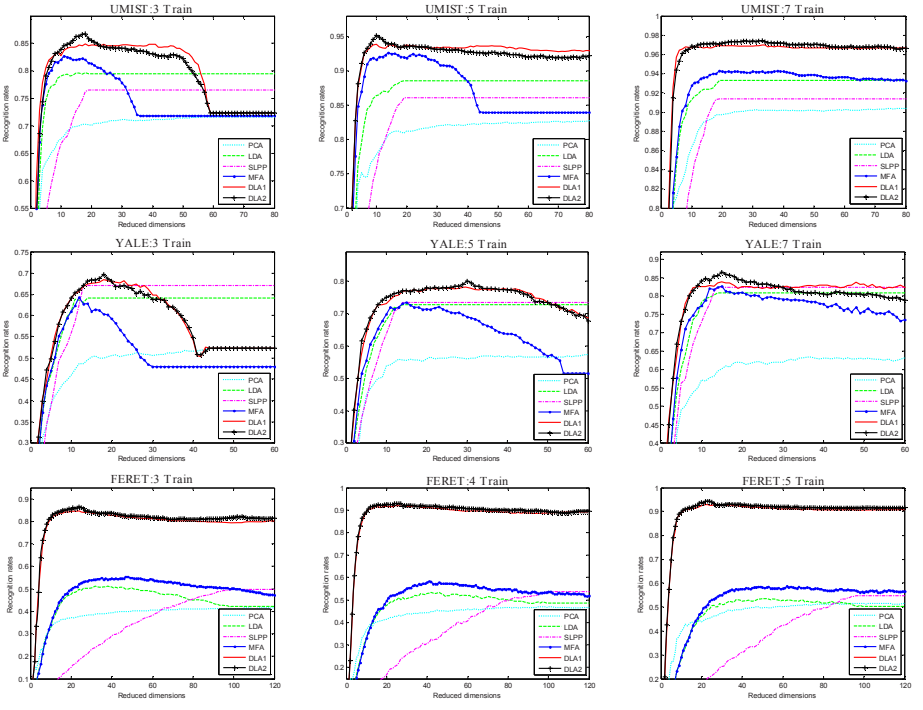


Fig. 4. Recognition rate vs. dimensionality reduction on three databases

Table 2. Best recognition rates (%) on FERET. The first numbers in the parentheses are the subspace dimensions, the second are the percent of energies retained in the PCA subspace.

	LDA	SLPP	MFA
3	78.03(17, 96%)	78.03(17, 96%)	78.95(21, 95%)
4	87.17(15, 96%)	87.17(15, 96%)	88.40(15, 94%)
5	91.85(21, 96%)	91.85(21, 96%)	92.35(19, 95%)

subspace dimensions and selecting the best one in LDA, SLPP and MFA. As shown in Table 2, although the performances of LDA, SLPP and MFA are significantly improved, DLA1 and DLA2 are still preponderant.

4.3 Building Patches

In this subsection, we study effects of k_1 and k_2 in DLA by setting $t = \infty$ in Eq. (7), based on the UMIST database with p (= 7) samples for each class in the training stage. The reduced dimension in experiments is fixed at 33. By varying k_1 from 1 to $p - 1$ (= 6) and k_2 from 0 to $N - p$ (= 133) simultaneously, the

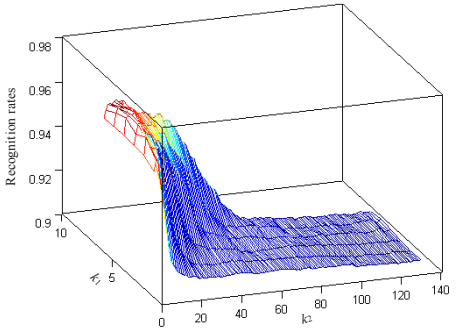


Fig. 5. Recognition rate vs. k_1 and k_2

recognition rate surface can be obtained as shown in Figure 5. In this figure, there is a peak which corresponds to $k_1 = 4$ and $k_2 = 5$.

In this experiment, optimal parameters k_1 and k_2 for classification can be obtained for patch building. It reveals that the local patch built by neighborhood can characterize not only the intrinsic geometry but also the discriminability better than the global structure.

4.4 Semi-supervised Experiments

We compare SDLA and DLA based on the UMIST database by setting $t = \infty$ in Eq. (7). The averaged recognition rates are obtained from ten different random runs. For each turn, $p\ (= 3, 5)$ samples with labels and $q\ (= 3, 5)$ samples without labels for each individual are selected randomly to train SDLA and DLA, and the left ones for each individual are used for testing. It is worth noting that q samples without labels have no effects in training DLA. Table 3 shows unlabeled samples are helpful to improve recognition rates.

Table 3. Recognition rates (%) of DLA and SDLA on UMIST. The numbers in the parentheses are the subspace dimensions.

		3 labeled	5 labeled
3 unlabeled	DLA	86.15 (15)	92.77 (11)
	SDLA	**87.69** (13)	**95.42** (22)
5 unlabeled	DLA	85.78 (27)	92.53 (11)
	SDLA	**88.19** (11)	**95.73** (30)

4.5 Discussions

Based on the experimental results reported in Section 4.2-4.4, we have the following observations:

1. DLA focuses on local patches; implements sample weighting for each part optimization; and avoids the matrix singularity problem. Therefore, it works better than PCA, LDA, SLPP, and MFA;

2. In experiments on building patches, by setting $k_1 = 6$ and $k_2 = 133$, DLA is similar to LDA because the global structure is considered. With this setting, DLA ignores the local geometry and performs poor. Thus, by setting k_1 and k_2 suitably, DLA can capture both the local geometry and the discriminative information of samples; and
3. Though analyses on SDLA, we can see that, although the unlabeled samples have no discriminative information, they are valuable to improve recognition rates by enhancing the local geometry of all samples.

5 Conclusions

In this paper, we have proposed a new linear dimensionality reduction algorithm, termed Discriminative Locality Alignment (DLA). The algorithm focuses on the local patch of every sample in a training set; implements the sample weighting by *margin degree*, a measure of the importance of each sample for classification; and never computes the inverse of a matrix. Advantages of DLA are that it distinguishes the contribution of each sample for discriminative subspace selection; overcomes the nonlinearity of the distribution of samples; preserves discriminative information over local patches; and avoids the matrix singularity problem. Experimental results have demonstrated the effectiveness of DLA by comparing with representative dimensionality reduction algorithms, e.g., PCA, LDA, SLPP, and MFA. An additional contribution is that we have also developed semi-supervised DLA (SDLA), which considers not only the labeled but also the unlabeled samples. Experiments have shown that SDLA performs better than DLA. It is worth emphasizing that the proposed DLA and SDLA algorithms can also be utilized to other interesting applications, e.g., pose estimation [17] emotion recognition [13], and 3D face modeling [12].

Acknowledgements

The work was partially supported by Hong Kong Research Grants Council General Research Fund (No. 528708), National Science Foundation of China (No. 60675023 and 60703037) and China 863 High Tech. Plan (No. 2007AA01Z164).

References

1. Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection. IEEE Trans. Pattern Analysis and Machine Intelligence 19(7), 711–720 (1997)
2. Belkin, M., Niyogi, P.: Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. Neural Information Processing Systems 14, 585–591
3. Belkin, M., Niyogi, P., Sindhwani, V.: On Manifold Regularization. In: Proc. Int'l Workshop on Artificial Intelligence and Statistics (2005)
4. Cai, D., He, X., Han, J.: Using Graph Model for Face Analysis. Technical report, Computer Science Department, UIUC, UIUCDCS-R-2005-2636 (2005)

5. Chen, L.F., Liao, H.Y., Ko, M.T., Lin, J.C., Yu, G.J.: A New LDA-based Face Recognition System Which Can Solve the Small Sample Size Problem. Pattern Recognition 33(10), 1713–1726 (2000)
6. Fisher, R.A.: The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics 7, 179–188 (1936)
7. Graham, D.B., Allinson, N.M.: Characterizing Virtual Eigensignatures for General Purpose Face Recognition. In: Face Recognition: From Theory to Applications. NATO ASI Series F, Computer and Systems Science, vol. 163, pp. 446–456 (2006)
8. He, X., Niyogi, P.: Locality Preserving Projections. In: Advances in Neural Information Processing Systems, vol. 16 (2004)
9. Hotelling, H.: Analysis of A Complex of Statistical Variables into Principal Components. Journal of Educational Psychology 24, 417–441 (1933)
10. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET Evaluation Methodology for Face-Recognition Algorithms. IEEE Trans. Pattern Analysis and Machine Intelligence 22(10), 1090–1104 (2000)
11. Roweis, S.T., Saul, L.K.: Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science 290, 2323–2326 (2000)
12. Song, M., Dong, Z., Theobalt, C., Wang, H., Liu, Z., Seidel, H.-P.: A Generic Framework for Efficient 2-D and 3-D Facial Expression Analogy. IEEE Trans. Multimedia 9(7), 1384–1395 (2007)
13. Song, M., You, M., Li, N., Chen, C.: A robust multimodal approach for emotion recognition. Neurocomputing 7(10-12), 1913–1920 (2008)
14. Tao, D., Li, X., Wu, X., Maybank, S.: Geometric Mean for Subspace Selection in Multiclass Classification. IEEE Trans. Pattern Analysis and Machine Intelligence 30 (2008)
15. Turk, M., Pentland, A.: Face Recognition using Eigenfaces. In: Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, pp. 586–591 (1991)
16. Yan, S., Xu, D., Zhang, B., Zhang, H.J., Yang, Q., Lin, S.: Graph Embedding and Extensions: A General Framework for Dimensionality Reduction. IEEE Trans. Pattern Analysis and Machine Intelligence 29(1), 40–51 (2007)
17. Yan, S., Wang, H., Fu, Y., Yan, J., Tang, X., Huang, T.: Synchronized Submanifold Embedding for Person-Independent Pose Estimation and Beyond. IEEE Trans. Image Processing (2008)
18. Yu, H., Yang, J.: A Direct LDA Algorithm for High-dimensional Data with Application to Face Recognition. Pattern Recognition 34(12), 2067–2070 (2001)
19. Zhang, Z., Zha, H.: Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment. SIAM J. Scientific Computing 26(1), 313–338 (2005)
20. Zhao, D., Lin, Z., Tang, X.: Laplacian PCA and Its Applications. In: Proc. IEEE Int'l Conf. Computer Vision, pp. 1–8 (2007)
21. Zhang, T., Tao, D., Li, X., Yang, J.: A Unifying Framework for Spectral Analysis based Dimensionality Reduction. In: Proc. Int'l J. Conf. Neural Networks (2008)
22. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised Learning using Gaussian Fields and Harmonic Functions. In: Proc. Int'l Conf. Machine Learning (2003)

Efficient Dense Scene Flow from Sparse or Dense Stereo Data

Andreas Wedel[1,2], Clemens Rabe[1], Tobi Vaudrey[3],
Thomas Brox[4], Uwe Franke[1], and Daniel Cremers[2]

[1] Daimler Group Research
firstname.lastname@daimler.com
[2] University of Bonn
dcremers@cs.uni-bonn.de
[3] University of Auckland
t.vaudrey@auckland.ac.nz
[4] University of Dresden
brox@inf.tu-dresden.de

Abstract. This paper presents a technique for estimating the three-dimensional velocity vector field that describes the motion of each visible scene point (scene flow). The technique presented uses two consecutive image pairs from a stereo sequence. The main contribution is to decouple the position and velocity estimation steps, and to estimate dense velocities using a variational approach. We enforce the scene flow to yield consistent displacement vectors in the left and right images. The decoupling strategy has two main advantages: Firstly, we are independent in choosing a disparity estimation technique, which can yield either sparse or dense correspondences, and secondly, we can achieve frame rates of 5 fps on standard consumer hardware. The approach provides dense velocity estimates with accurate results at distances up to 50 meters.

1 Introduction

A very important feature to extract from a moving scene is the velocity of visible objects. In the scope of the human nerve system such perception of motion is referred to as kinaesthesia. The motion in 3D space is called *scene flow* and can be described by a three-dimensional velocity field.

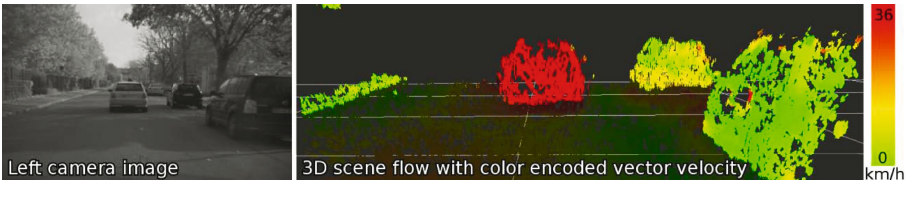


Fig. 1. Scene flow example. Despite similar distance from the viewer, the moving car (red) can be clearly distinguished from the parked vehicles (green).

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 739–751, 2008.

With images from a single camera, scene flow computation is clearly underdetermined, due to the projection on to the image plane. Even if the camera is moving, there arise ambiguities between the camera motion and the motion of objects in the scene. These ambiguities are largely resolved when using a second camera. Ambiguities only remain due to missing structure in local parts of the image. In 2D motion estimation, this is known as the aperture problem. A common way to deal with this problem is by using a variational framework (e.g. [6]), which includes a smoothness assumption on the velocity field. This allows for dense motion estimation, despite missing structure in parts of the image.

In the case of 3D motion estimation, we can also make use of a variational technique in order to achieve dense estimates (as done in [7]). However, it should be clear that only the motion of visible parts of the scene can be estimated. For our purposes, we refer to dense scene flow as the 3D velocity vector at each 3D point that can be seen by both cameras.

Scene flow estimation, with known camera parameters, involves estimating the 3D velocity in consecutive stereo frames, and also the disparity needed to calculate the absolute position of the world point. In this paper, we suggest performing the velocity estimation and the disparity estimation separately, while still ensuring consistency of all involved frames. The decoupling of depth (3D position) and motion (3D scene flow) estimation implies that we do not enforce depth consistency between t and t+1. While splitting the problem into two subproblems might look unfavourable at a first glance, it only affects the accuracy of the disparity estimate and has two important advantages.

Firstly, the challenges in motion estimation and disparity estimation are very different. With disparity estimation, thanks to the epipolar constraint, only a scalar field needs to be estimated. This enables the use of efficient global optimisation methods, such as dynamic programming or graph-cuts, to establish point correspondences. Optical flow estimation, on the other hand, requires the estimation of a vector field, which rules out such global optimisation strategies. Additionally, motion vectors are usually much smaller in magnitude than disparities. With optical flow, occlusion handling is less important than the sub-pixel accuracy provided by variational methods. Splitting scene flow computation into the estimation sub-problems, disparity and optical flow, allows us to choose the optimal technique for each task.

Secondly, the two sub-problems can be solved more efficiently than the joint problem. This allows for real-time computation of scene flow, with a frame rate of 5 fps on QVGA images (320×240 pixel). This is about 500 times faster compared to the recent technique for joint scene flow computation in [7]. Nevertheless, we achieve accuracy that is at least as good as the joint estimation method.

1.1 Related Work

2D motion vectors are obtained by optical flow estimation techniques. There are dense as well as sparse techniques. Sparse optical flow techniques, such as KLT tracking [16], usually perform some kind of feature tracking and are preferred in time-critical applications, due to computational benefits. Dense optical

flow is mostly provided by variational models based on the method of Horn and Schunck [6]. Local variational optimisation is used to minimise an energy function that assumes constant pixel intensities and a smooth flow field. The basic framework of Horn and Schunck has been improved over time to cope with discontinuities in the flow field, and to obtain robust solutions with the presence of outliers in image intensities [9]. Furthermore, larger displacements can be estimated thanks to image warping and non-linearised model equations [1,9]. Currently, variational techniques yield the most accurate optical flow in the literature. Real-time methods have been proposed in [2,18].

Scene flow involves an additional disparity estimation problem, as well as the task to estimate the change of disparity over time. The work in [11] introduced scene flow as a joint motion and disparity estimation method. The succeeding works in [7,10,19] presented energy minimisation frameworks including regularisation constraints to provide dense scene flow. Other dense scene flow algorithms have been presented in multiple camera set-ups [13,17]. However, these allow for non-consistent flow fields in single image pairs. At this point it is worth noting that, although we separate the problems of disparity estimation and motion estimation, the method still involves a coupling of these two tasks, as the optical flow is enforced to be consistent with the computed disparities.

None of the above approaches run in real-time, giving best performances in the scale of minutes. Real-time scene flow algorithms, such as the one presented in [14], provide only sparse results both for the disparity and the velocity estimates. The work in [8] presents a probabilistic scene flow algorithm with computation times in the range of seconds, but yielding only integer pixel-accurate results. In contrast, the method we present in this paper provides sub-pixel accurate scene flow in real-time for reasonable image sizes. Furthermore, in combination with a dense disparity map the scene flow field is dense.

2 Scene Flow and Its Constraints on Image Motion

2.1 Stereo Computation

Assume we are given a pair of stereo images. Normal stereo epipolar geometry is assumed, such that pixel rows y for the left and right images coincide. In practice, this is achieved by a rectification step, which warps the images according to the known intrinsic and relative extrinsic configuration of the two involved cameras [4,12]. In addition, the principal points of both images are rearranged, such that they lie on the same image coordinates (x_0, y_0). A world point (X, Y, Z) given in the camera coordinate system is projected onto the image point (x, y) in the left image, and the image point $(x + d, y)$ in the right image according to

$$\begin{pmatrix} x \\ y \\ d \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} X f_x \\ Y f_y \\ -b f_x \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \\ 0 \end{pmatrix} \tag{1}$$

with the focal lengths f_x and f_y, in pixels, and the distance b as baseline, in metres, between the two camera projection centres. The disparity value d

therefore encodes the difference in the x-coordinate of an image correspondence. With known camera parameters, the position of a world point can easily be recovered from an (x, y, d) measurement according to Equation (1).

The goal of any stereo algorithm is to determine the disparity d, in order to reconstruct the 3D scene. This is accomplished by either matching a small window from the the left image to an area in the right image, or by calculating a globally consistent solution, using energy minimisation techniques. The issue with the presented scene flow framework is that we can employ any stereo algorithm. In Section 5 this is demonstrated, as we show results with various sparse and dense stereo algorithms.

2.2 Constraints on Image Motion

Assume we are given two consecutive pairs of stereo images at time t and $t+1$. Analogous to the optical flow field, the scene flow field is the projection of the three-dimensional motion field. It provides for each pixel a change in the image space (u, v, d') between the two stereo image pairs, where u and v is the change in image x and y respectively, and d' is the change in disparity, all in pixels. The three-dimensional velocity field can be reconstructed, if both the image measurement (x, y, d) and its change (u, v, d') are known. Leaving the estimation of d to an arbitrary stereo algorithm, we will now derive the constraints for estimating (u, v, d').

Let $I(x, y, t)^l, I(x, y, t)^r$ be the intensity value of the left and right image, respectively, at time t and image position (x, y). Using Equation (1), a correspondence between the left and right stereo image at time t can be represented as $I(x, y, t)^l$ and $I(x + d, y, t)^r$. Since the flow in y-direction has to be equal in both images due to rectification, the constraints for the optical flow in the left and right images are:

$$I(x, y, t)^l = I(x + u, y + v, t + 1)^l \tag{2}$$

$$I(x + d, y, t)^r = I(x + d + d' + u, y + v, t + 1)^r \tag{3}$$

If the disparity d is known, the right image at time t is redundant for solving the scene flow problem, because $I(x, y, t)^l = I(x + d, y, t)^r$. In practice, $I(x, y, t)^l = I(x + d, y, t)^r$ does not hold exactly even for perfect d, since we have illumination changes between two different cameras. Therefore, we use the optical flow constraints for the left and right camera images separately, as stated in the above formulas.

Calculating optical flow in the left and right image separately, we could derive the disparity change $d' = u^r - u^l$, where u^r and u^l denote the estimated flow fields in the left and right image, respectively. However, we introduce an additional constraint, enforcing consistency of the left and right image at $t + 1$:

$$I(x + u, y + v, t + 1)^l - I(x + d + d' + u, y + v, t + 1)^r = 0 \tag{4}$$

3 A Variational Framework for Scene Flow

Scene flow estimates according to the constraints formulated in Section 2 can be computed in a variational framework by minimising an energy functional consisting of a constraint deviation term and a smoothness term that enforces smooth and dense scene flow:

$$E(u, v, d') = E_{\text{Data}}(u, v, d') + E_{\text{Smooth}}(u, v, d') \tag{5}$$

Integrating the constraints from Section 2 over the image domain Ω, we obtain the following data term:

$$\begin{aligned} E_{\text{Data}} &= \int_\Omega \Psi\left(\left(I(x+u, y+v, t+1)^l - I(x, y, t)^l\right)^2\right) dxdy \\ &+ \int_\Omega c(x,y)\,\Psi\left(\left(I(x_d+d'+u, y+v, t+1)^r - I(x_d, y, t)^r\right)^2\right) dxdy \\ &+ \int_\Omega c(x,y)\,\Psi\left(\left(I(x_d+d'+u, y+v, t+1)^r - I(x+u, y+v, t+1)^l\right)^2\right) dxdy \end{aligned} \tag{6}$$

where $\Psi(s^2) = \sqrt{s^2 + \varepsilon}$ ($\varepsilon = 0.0001$) denotes a robust function that compensates for outliers [1], and $x_d := x + d$ for simpler notation. The indicator function $c(x, y) : \Omega \to \{0, 1\}$ returns 0 if there is no disparity known at (x, y). This can be due to a point seen only in one camera (occlusion) or due to a non-dense stereo method. Otherwise $c(x, y)$ returns 1. The first term in Equation (6) imposes the brightness constancy for the left images, the second one for the right images, and the third one assures consistency of the estimated motion between left and right images.

The smoothness term penalises local deviations in the scene flow components and employs the same robust function as the data term in order to deal with discontinuities in the velocity field:

$$E_{\text{Smooth}} = \int_\Omega \Psi\left(\lambda|\nabla u|^2 + \lambda|\nabla v|^2 + \gamma|\nabla d'|^2\right) dxdy \tag{7}$$

where $\nabla = (\partial/\partial x, \partial/\partial y)$. The parameters λ and γ regulate the importance of the smoothness constraint, weighting for optic flow and disparity change respectively. Interestingly, due to the fill-in effect of the above regularizer, the proposed variational formulation provides dense scene flow estimates (u, v, d'), even if the disparity d is non-dense.

4 Minimisation of the Energy

For minimising the above energy we compute its Euler-Lagrange equations:

$$\begin{aligned} &\Psi'((I_z^l)^2) I_z^l I_x^l + c\,\Psi'((I_z^r)^2) I_z^r I_x^r + c\,\Psi'((I_z^d)^2) I_z^d I_x^r \\ &\quad -\lambda \operatorname{div}\left(\Psi'(\lambda|\nabla u|^2 + \lambda|\nabla v|^2 + \gamma|\nabla d'|^2)\nabla u\right) = 0 \end{aligned} \tag{8}$$

$$\begin{aligned}&\Psi'((I_z^l)^2)I_z^l I_y^l + c\Psi'((I_z^r)^2)I_z^r I_y^r + c\Psi'((I_z^d)^2)I_z^d I_y^r\\&\quad -\lambda \operatorname{div}\left(\Psi'(\lambda|\nabla u|^2 + \lambda|\nabla v|^2 + \gamma|\nabla d'|^2)\nabla v\right) = 0\end{aligned} \tag{9}$$

$$\begin{aligned}&c\Psi'((I_z^r)^2)I_z^r I_x^r + c\Psi'((I_z^d)^2)I_z^d I_x^r\\&\quad -\gamma \operatorname{div}\left(\Psi'(\lambda|\nabla u|^2 + \lambda|\nabla v|^2 + \gamma|\nabla d'|^2)\nabla d'\right) = 0\end{aligned} \tag{10}$$

where $\Psi'(s^2)$ denotes the derivative of Ψ with respect to s^2. We define $I_z^l := I(x+u, y+v, t+1)^l - I(x, y, t)^l$, $I_z^r := I(x_d + d' + u, y+v, t+1)^r - I(x_d, y, t)^r$, and $I_z^d := I(x_d + d' + u, y+v, t+1)^r - I(x+u, y+v, t+1)^l$, and subscripts x and y denote the respective partial derivatives of $I(x+u, y+v, t+1)^l$ and $I(x_d + d' + u, y+v, t+1)^r$.

These equations are non-linear in the unknowns, so we stick to the strategy of two nested fixed point iteration loops as suggested in [1]. This comes down to a warping scheme as also employed in [9]. The basic idea is to have an outer fixed point iteration loop that comprises linearisation of the I_z expressions. In each iteration, an increment of the unknowns is estimated and the second image is then warped according to the new estimate. The warping is combined with a coarse-to-fine strategy, where we work with down-sampled images that are successively refined with the number of iterations. Since we are interested in real-time estimates, we use only 4 scales with 2 outer fixed point iterations at each scale.

In the present case, we have the following linearisation, where k denotes the iteration index. We start the iterations with $(u^0, v^0, d'^0)^\top = (0, 0, 0)^\top$:

$$I(x+u^k+\delta u^k, y+v^k+\delta v^k, t+1)^l \approx I(x+u^k, y+v^k, t+1)^l + \delta u^k I_x^l + \delta v^k I_y^l \tag{11}$$

$$\begin{aligned}&I(x_d + d'^k + \delta d'^k + u^k + \delta u^k, y + v^k + \delta v^k, t+1)^r\\&\quad \approx I(x_d + d'^k + u^k, y + v^k, t+1)^r + \delta u^k I_{x_d}^r + \delta d'^k I_{x_d}^r + \delta v^k I_y^r\end{aligned} \tag{12}$$

From these expressions we can derive linearised versions of I_z^*. The remaining non-linearity in the Euler-Lagrange equations is due to the robust function. In the inner fixed point iteration loop the Ψ' expressions are kept constant and are recomputed after each iteration. This finally leads to the following linear equations:

$$\begin{aligned}0 = {}&\Psi'((I_z^{l,k+1})^2)(I_z^{l,k} + I_x^{l,k}\delta u^k + I_y^{l,k}\delta v^k)I_x^{l,k}\\&+c\Psi'((I_z^{r,k+1})^2)(I_z^{r,k} + I_x^{r,k}(\delta u^k + \delta d'^k) + I_y^{r,k}\delta v^k)I_x^{r,k}\\&-\lambda \operatorname{div}\left(\Psi'(\lambda|\nabla u^{k+1}|^2 + \lambda|\nabla v^{k+1}|^2 + \gamma|\nabla d'^{k+1}|^2)\nabla(u^k + \delta u^k)\right)\end{aligned} \tag{13}$$

$$\begin{aligned}0 = {}&\Psi'((I_z^{l,k+1})^2)(I_z^{l,k} + I_x^{l,k}\delta u^k + I_y^{l,k}\delta v^k)I_y^{l,k}\\&+c\Psi'((I_z^{r,k+1})^2)(I_z^{r,k} + I_x^{r,k}(\delta u^k + \delta d'^k) + I_y^{r,k}\delta v^k)I_y^{r,k}\\&-\lambda \operatorname{div}\left(\Psi'(\lambda|\nabla u^{k+1}|^2 + \lambda|\nabla v^{k+1}|^2 + \gamma|\nabla d'^{k+1}|^2)\nabla(v^k + \delta v^k)\right)\end{aligned} \tag{14}$$

$$\begin{aligned}0 = {}&c\Psi'((I_z^{r,k+1})^2)(I_z^{r,k} + I_x^{r,k}(\delta u^k + \delta d'^k) + I_y^{r,k}\delta v^k)I_x^{r,k}\\&+c\Psi'((I_z^{d,k+1})^2)(I_z^{d,k} + I_x^{r,k}\delta d'^k)I_x^{r,k}\\&-\gamma \operatorname{div}\left(\Psi'(\lambda|\nabla u^{k+1}|^2 + \lambda|\nabla v^{k+1}|^2 + \gamma|\nabla d'^{k+1}|^2)\nabla(d'^k + \delta d'^k)\right)\end{aligned} \tag{15}$$

where we omitted the iteration index of the inner fixed point iteration loop to keep the notation uncluttered. Expressions with iteration index $k+1$ are

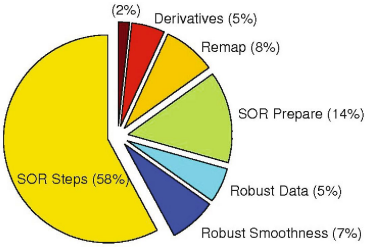


Image Size (pixels)	SOR Steps per Warp	Warps per Level	Total Time
640 × 480	45	2	975 ms
512 × 512	20	3	756 ms
320 × 240	45	2	205 ms

Fig. 2. Break down of computational time for our algorithm (3.0GHz Intel®Core$^{\text{TM}}$2). The pie graph shows the time distribution for the 640 × 480 images. The real-time applicability of the algorithm for image sizes of (320 × 240) is indicated in the table.

computed using the current increments $\delta u^k, \delta v^k, \delta d'^k$. We see that some terms of the original Euler-Lagrange equations have vanished as we have made use of $I(x_d, y, t)^r = I(x, y, t)^l$ in the linearised third constraint (Equation 4). After discretisation, the corresponding linear system is solved via successive over-relaxation. It is worth noting that, for efficiency reasons, it is advantageous to update the Ψ' after a few iterations of SOR.The shares of computation time taken by the different operations are shown in Figure 2.

5 Results

To assess the quality of our scene flow algorithm, it was tested on synthetic sequences, where the ground truth is known[1]. In a second set of experiments, we used real images to demonstrate the accuracy and practicality of our algorithms under real world conditions.

Synthetic scenes. The first ground truth experiment is the *rotating sphere* sequence from [7] depicted in Figure 3. In this sequence the spotty sphere rotates around its y-axis to the left, while the two hemispheres of the sphere rotate in opposing vertical directions. The resolution is 512 × 512 pixels.
We tested the scene flow method together with four different stereo algorithms: semi-global matching (SGM [5]), SGM with hole filling (favours smaller disparities), correlation pyramid stereo [3], and an integer accurate census-based stereo algorithm [15]. The ground truth disparity was also used for comparison. For each stereo algorithm, we calculated the absolute angular error (AAE) and the root mean square (RMS) error

$$RMS_{u,v,d,d'} := \sqrt{\frac{1}{n}\sum_{\Omega} \|(u_i, v_i, d_i, d'_i)^\top - (u_i^*, v_i^*, d_i^*, d_i'^*)^\top\|^2} \qquad (16)$$

where a superscript* denotes the ground truth solution and n is the number of pixels. In our notation for RMS, if a subscript is omitted, then both the

[1] The authors would like to thank Huguet *et al.* for providing their *sphere* scene.

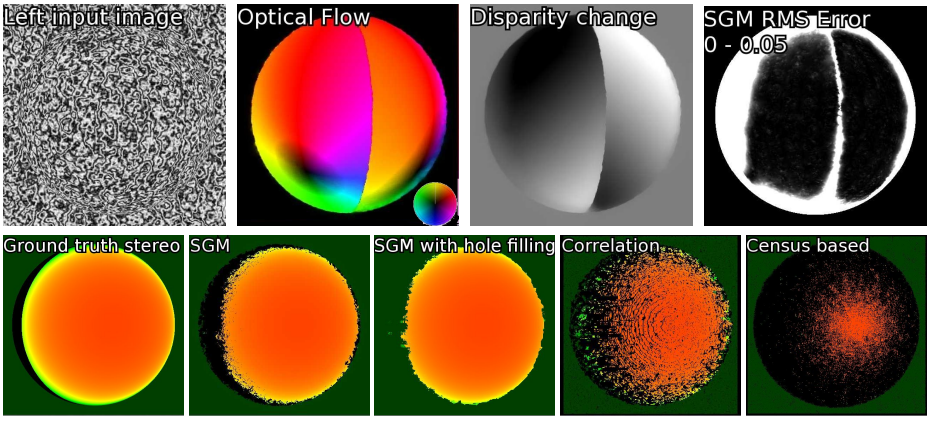


Fig. 3. Ground truth test: *rotating sphere*. Quantitative results are shown in Table 1. **Top:** Optical flow and disparity change are computed on the basis of SGM stereo [5]. Colour encodes the direction of the optical flow (key in bottom right), intensity its magnitude. Disparity change is encoded from black (increasing) to white (decreasing). Bright parts of the RMS figure indicate high $RMS_{u,v,d'}$ error values of the computed scene flow. **Bottom:** Disparity images are colour encoded green to orange (low to high). Black areas indicate missing disparity estimates or occluded areas.

respective ground truth and estimated value are set to zero. The errors were calculated in using two types of Ω: firstly, calculating statistics over all non-occluded areas, and secondly calculating over the whole sphere. As in [7], pixels from the background were not included in the statistics.

The smoothing parameters were set to $\lambda = 0.2$ and $\gamma = 2$. We used 60 SOR iterations at each pyramid level, resulting in an average runtime of 756 milliseconds. Additionally, we let the algorithm run to convergence (change in $RMS \leq \varepsilon$) for better accuracy without changing λ and γ; this increased the computational time to around 3 seconds.

The resulting summary can be seen in Table 1. We achieve lower errors than the Huguet *et al.* method, when we let the method converge. Particularly, the RMS error of the scene flow is much smaller and we are still considerably faster. This is explained by the higher flexibility in choosing the disparity estimation method. Furthermore, we achieve real-time performance with little loss in accuracy. The table shows that SGM with hole filling yields inferior results than the other stereo methods. This is due to false disparity measurements in the occluded area. It is better to feed the sparse measurements of SGM to the variational framework, which yields dense estimates as well, but with higher accuracy. SGM was used in the remainder of the results section, as it gives the best results and is available on dedicated hardware without any extra computational cost.

In a second ground truth example we use a Povray-rendered traffic scene. The scene includes common features such as mixture of high and low textured areas on the ground plane and background, occlusions, reflections, and transparency of car windscreens. The vehicle in front of the camera (and its shadow) moves

Table 1. Root mean square (pixels) and average angular error (degrees) for scene flow of the *rotating sphere* sequence. Various stereo algorithms are used as input for our scene flow estimation, generating varying results. A * denotes running until convergence. SGM (highlighted) is the best solution for its speed to accuracy ratio. "Flow Only" does not include stereo correspondences, thus calculates 2D optical flow only. For the evaluation we used the formula $AAE_{u,v} := \frac{1}{n} \int_{\Omega} \arctan \frac{uv^* - u^* v}{uu^* + vv^*}$ to calculate the error to the ground truth flow (u^*, v^*) as used in [7].

Stereo Algorithm	RMS_d (density)	Without occluded areas			With occluded areas		
		$RMS_{u,v}$	$RMS_{u,v,d'}$	$AAE_{u,v}$	$RMS_{u,v}$	$RMS_{u,v,d'}$	$AAE_{u,v}$
Huguet *et al.* [7]	3.8 (100%)	0.37	0.83	1.24	0.69	2.51	1.75
Flow Only	$d' = 0$ for	0.34	1.46	1.26	0.67	2.85	1.72
Flow Only*	evaluation	0.30	1.46	0.95	0.64	2.85	1.36
Ground truth		0.33	0.58	1.25	0.67	2.40	1.78
Ground truth*		0.31	0.56	0.91	0.65	2.40	1.40
SGM [5]	2.9 (87%)	0.35	0.64	1.33	0.66	2.45	1.82
SGM*		0.34	0.63	1.04	0.66	2.45	1.50
Fill-SGM	10.9 (100%)	0.43	0.75	2.18	0.77	2.55	2.99
Fill-SGM*		0.45	0.76	1.99	0.77	2.55	2.76
Correlation [3]	2.6 (43%)	0.34	0.75	1.31	0.67	2.51	1.84
Correlation*		0.33	0.73	1.02	0.65	2.50	1.52
Census based [15]	7.8 (16%)	0.36	1.08	1.30	0.67	2.65	1.75
Census based*		0.32	1.14	1.01	0.65	2.68	1.43

straight ahead. There are vehicles entering the main road from the left and the right. The camera system is also moving orthogonal to the image plane. We calculated the $RMS_{u,v,d'}$ error and the 4D angular error defined by:

$$AAE_{4D} := \frac{1}{n} \sum_{\Omega} \arccos \left(\frac{uu^* + vv^* + d'd'^* + 1}{\sqrt{(u^2 + v^2 + d'^2 + 1)\,((u^*)^2 + (v^*)^2 + (d'^*)^2 + 1)}} \right)$$

Results are shown in Figures 4 and 5. They compare favourably to the results obtained when running the code from [7]. The average $RMS_{u,v,d'}$ error for the whole sequence (subregion as in Figure 4) was 0.64 px and the 4D angular error was 3.01°. The sequence will be made publicly available in the internet to allow evaluation of future scene flow algorithms (*http://citr.auckland.ac.nz/6D/*).

Real world scenes. Figure 6 and Figure 7 show scene flow results in real world scenes with a moving camera. A result video of the scene shown in Figure 1 is included in the supplemental material. Ego motion of the camera is known from vehicle odometry and compensated in the depicted results.

Figure 6 shows an image from a sequence where the ego-vehicle is driving past a bicyclist. The depicted scene flow shows that most parts of the scene, including the vehicle stopping at the traffic lights, are correctly estimated as stationary. Only the bicyclist is moving and its motion is accurately estimated.

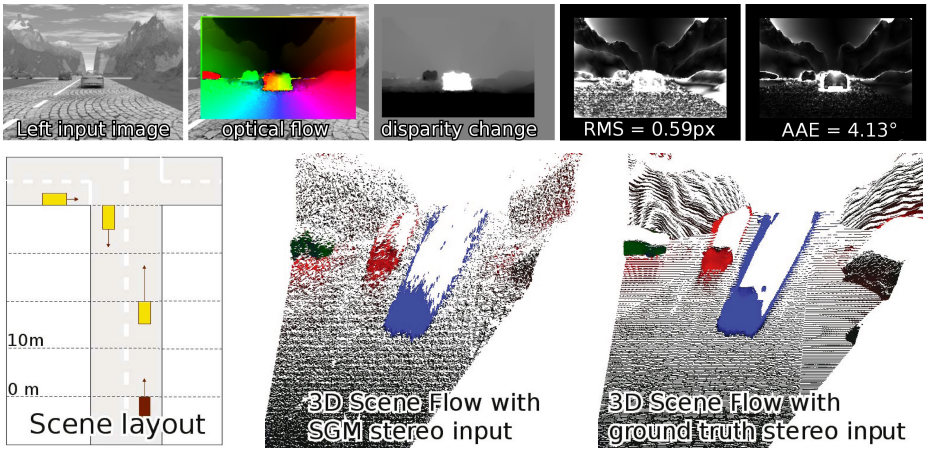


Fig. 4. Povray-rendered traffic scene (Frame 11). **Top:** Colour encodes direction (border = direction key) and intensity the magnitude of the optical flow vectors. Brighter areas in the error images denote larger errors. For comparison, running the code from [7] generates an RMS error of 0.91px and AAE of 6.83°. **Bottom right:** 3D views of the scene flow vectors. Colour encodes their direction and brightness their magnitude (black = stationary). The results from the scene are clipped at a distance of 100m. Accurate results are obtained even at greater distances.

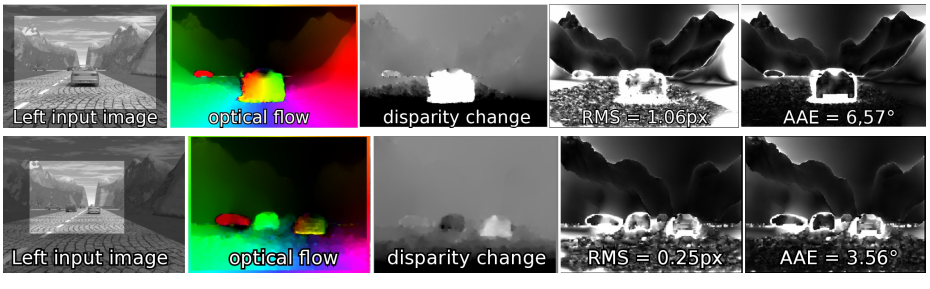


Fig. 5. More frames from the traffic scene in Figure 4. The top row highlights the problems such as transparency of the windshield, reflectance, and moving shadows. The bottom row demonstrates that we still maintain accuracy at distances of 50 m.

Figure 7 shows results from a scene where a person runs from behind a parked vehicle. The ego-vehicle is driving forward at 30 km/h and turning to the left. The measurements on the ground plane and in the background are not shown to focus visual attention on the person. The results show that points on the parked vehicle are estimated as stationary, where as points on the person are registered as moving. The accurate motion results can be well observed for the person's legs, where the different velocities of each leg are well estimated.

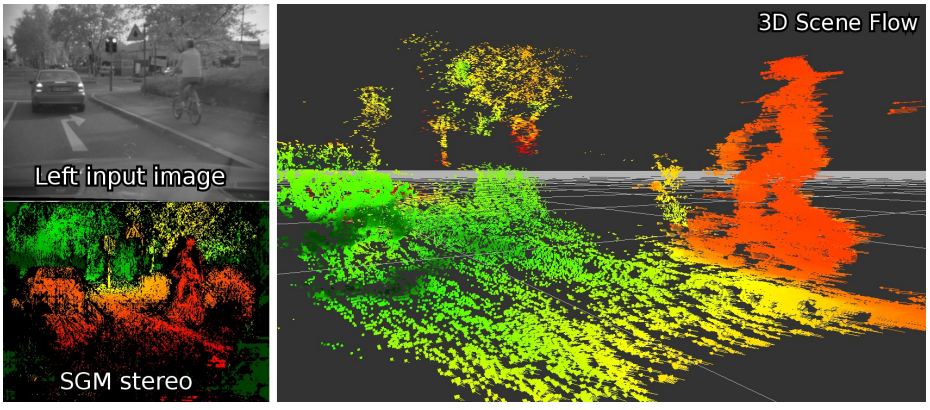


Fig. 6. Dense scene flow in a traffic scene. The colour in the lower left image encodes distance from red to green (close to far); the colour in the scene flow image (right) shows vector lengths after ego-motion compensation (green to red = 0 to $0.4m/s$). Only the cyclist is moving.

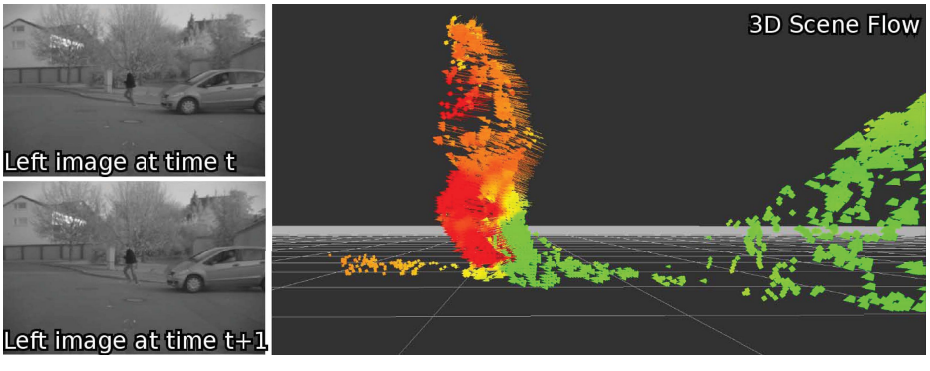


Fig. 7. Scene with a person running from behind a parked vehicle. The colour encoding is as in Figure 6.

6 Conclusions

We presented a variational framework for dense scene flow estimation, which is based on a decoupling of the disparity estimation from the velocity estimation, while enforcing consistent motion in all involved images. We showed that this strategy has two main advantages: Firstly, we can choose optimal methods for estimating both disparity and velocity. In particular, we can combine occlusion handling and global (combinatorial) optimisation for disparity estimation with dense, sub-pixel accurate velocity estimation. Secondly, for the first time, we obtain dense scene flow results very efficiently in real-time. We showed that the approach works well on both synthetic and real sequences, and that

it provides highly accurate velocity estimates, which compare favourably to the literature. Ongoing work will include temporal consistency by employing, for instance, Kalman filters. Another interesting aspect is the segmentation of the 3D velocity field.

References

1. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
2. Bruhn, A., Weickert, J., Kohlberger, T., Schnörr, C.: Discontinuity preserving computation of variational optic flow in real-time. In: ScaleSpace 2005, pp. 279–290 (2005)
3. Franke, U., Joos, A.: Real-time stereo vision for urban traffic scene understanding. In: Proc. IEEE Intelligent Vehicles Symposium, Dearborn, pp. 273–278 (2000)
4. Fusiello, A., Trucco, E., Verri, A.: A compact algorithm for rectification of stereo pairs. Machine Vision and Applications 12(1), 16–22 (2000)
5. Hirschmüller, H.: Stereo vision in structured environments by consistent semi-global matching. In: CVPR (2), pp. 2386–2393. IEEE Computer Society, Los Alamitos (2006)
6. Horn, B., Schunck, B.: Determining optical flow. Artificial Intelligence 17, 185–203 (1981)
7. Huguet, F., Devernay, F.: A variational method for scene flow estimation from stereo sequences. In: IEEE Eleventh International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil (October 2007)
8. Isard, M., MacCormick, J.: Dense motion and disparity estimation via loopy belief propagation. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006. LNCS, vol. 3852, pp. 32–41. Springer, Heidelberg (2006)
9. Mémin, E., Pérez, P.: Dense estimation and object-based segmentation of the optical flow with robust techniques. IEEE Transactions on Image Processing 7(5), 703–719 (1998)
10. Min, D., Sohn, K.: Edge-preserving simultaneous joint motion-disparity estimation. In: ICPR 2006: Proc. 18th International Conference on Pattern Recognition, Washington, DC, USA, pp. 74–77. IEEE Computer Society Press, Los Alamitos (2006)
11. Patras, I., Hendriks, E., Tziritas, G.: A joint motion/disparity estimation method for the construction of stereo interpolated images in stereoscopic image sequences. In: Proc. 3rd Annual Conference of the Advanced School for Computing and Imaging, Heijen, The Netherlands (1997)
12. Pollefeys, M., Koch, R., Gool, L.V.: A simple and efficient rectification method for general motion. In: IEEE International Conference on Computer Vision, pp. 496–501 (1999)
13. Pons, J.-P., Keriven, R., Faugeras, O.: Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. Int. J. Comput. Vision 72(2), 179–193 (2007)
14. Rabe, C., Franke, U., Gehrig, S.: Fast detection of moving objects in complex scenarios. In: Proc. IEEE Intelligent Vehicles Symposium, pp. 398–403 (June 2007)
15. Stein, F.: Efficient computation of optical flow using the census transform. In: Rasmussen, C.E., Bülthoff, H.H., Schölkopf, B., Giese, M.A. (eds.) DAGM 2004. LNCS, vol. 3175, pp. 79–86. Springer, Heidelberg (2004)

16. Tomasi, C., Kanade, T.: Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University (April 1991)
17. Vedula, S., Baker, S., Rander, P., Collins, R., Kanade, T.: Three-dimensional scene flow. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(3), 475–480 (2005)
18. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime tv-L^1 optical flow. In: Proc. DAGM (Pattern Recognition), pp. 214–223 (2007)
19. Zhang, Y., Kambhamettu, C.: On 3d scene flow and structure estimation. In: Proc. IEEE Conf. in Computer Vision and Pattern Recognition, vol. 2, p. 778. IEEE Computer Society Press, Los Alamitos (2001)

Integration of Multiview Stereo and Silhouettes Via Convex Functionals on Convex Domains

Kalin Kolev and Daniel Cremers

Department of Computer Science,
University of Bonn, Germany
{kolev,dcremers}@cs.uni-bonn.de

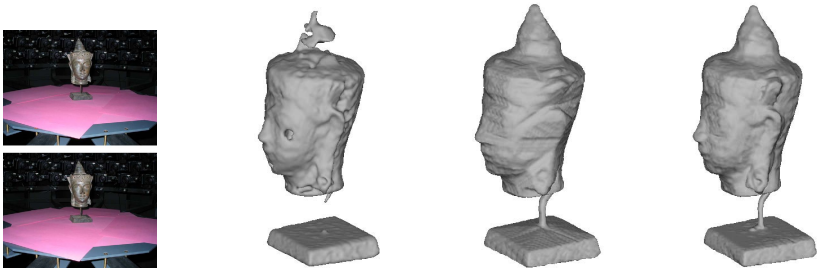
Abstract. We propose a convex framework for silhouette and stereo fusion in 3D reconstruction from multiple images. The key idea is to show that the reconstruction problem can be cast as one of minimizing a *convex* functional where the exact silhouette consistency is imposed as a convex constraint that restricts the domain of admissible functions. As a consequence, we can retain the original stereo-weighted surface area as a cost functional without heuristic modifications by balloon terms or other strategies, yet still obtain meaningful (nonempty) global minimizers. Compared to previous methods, the introduced approach does not depend on initialization and leads to a more robust numerical scheme by removing the bias near the visual hull boundary. We propose an efficient parallel implementation of this convex optimization problem on a graphics card. Based on a photoconsistency map and a set of image silhouettes we are therefore able to compute highly-accurate and silhouette-consistent reconstructions for challenging real-world data sets in less than one minute.

1 Introduction

Recovering three-dimensional geometrical structure from a series of calibrated images is among the fundamental problems in computer vision, with numerous applications in computer graphics, augmented reality, robot navigation and tracking. Among the multitude of existing methods for multiview reconstruction one can identify two major classes of approaches: shape from silhouettes and shape from stereo.

Historically, the first strategy for multiview 3D shape retrieval, dating back to the 1970's, has been to use the outlines of the imaged objects [1]. Most of these *shape from silhouettes* approaches aim at approximating the visual hull [2] of the observed solid. The visual hull is an outer approximation, constructed as the intersection of the visual cones associated with all image silhouettes. In the course of research, different shape representations have been proposed: volumetric [3], surface-based [4] and polyhedral [5]. Apart from shape representation, research has been also focused on the development of methods operating on raw image data instead of predetermined silhouettes. Most of them are based on an energy minimization framework allowing to impose regularization in the labeling process

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 752–765, 2008.



(a) (b) (c) (d)

Fig. 1. Silhouette and stereo integration. (a) Two of the input images. While stereo-based approaches (b) recover concavities but overcarve thin structures and areas of specular reflections, silhouette-based methods (c) reconstruct small-scale details captured by the silhouettes but fill indentations. In contrast, techniques for silhouette and stereo fusion (d) restore concave areas as well as fine geometric details.

[6]. The segmentation of each image is obtained through the evolution of a single surface in 3D rather than separate contours in 2D. As a result, such approaches exhibit considerable robustness to image noise and erroneous camera calibration.

The main drawback of silhouette-based approaches is their inability to reconstruct concavities, since these do not affect the silhouettes. *Stereo-based methods* capture such indentations by measuring photoconsistency of surface patches in space. The fundamental idea is that under the Lambertian assumption only points on the object's surface have a consistent appearance in the input images, while all other points project to incompatible image patches. The earliest algorithms use carving techniques to obtain a volumetric representation of the scene by repeatedly eroding inconsistent voxels [7]. They do not enforce the smoothness of the surface and this often results in rather noisy reconstructions. This drawback was overcome by energy minimization techniques [8,9,10], which typically aim at computing a weighted minimal surface, where the weights reflect the local photoconsistency.

Ideally, one would therefore like to combine both types of techniques in order to jointly achieve stereo and silhouette consistency (see Fig. 1). A simple strategy to fuse these complementary features is to use a visual hull (computed from silhouettes) as initialization for a stereo-based approach [11,12]. Firstly, this requires to constrain the solution space to avoid the empty set, secondly, the resulting reconstruction will generally not fulfill the silhouette constraint. Alternatively, one can unify both information sources in a single formulation. Two different techniques have been proposed to achieve this goal: One can integrate in the evolution silhouette-aligning forces [13,14,15,16], or one can use predetermined surface points [17,18,19] to impose exact silhouette constraints. Both strategies have their shortcomings. The first one could lead to a numerically unstable behavior and could introduce a bias near the visual hull boundary, while the second one requires premature decisions about voxel occupancy. To address these drawbacks [20] proposed a graph cut framework for silhouette and stereo

fusion. Unfortunately, the practical applicability of this method is limited due to its high memory requirements, which poses a severe restriction on the volume resolution. As a consequence, the development of robust and efficient schemes for silhouette and stereo integration remains an open challenge.

In this paper, we propose a novel mathematical framework for silhouette and stereo fusion in 3D reconstruction. The idea is to cast multiview stereovision as a *convex* variational problem where the exact silhouette consistency is imposed as a convex constraint that restricts the domain of admissible functions. Silhouette-consistent reconstructions are computed by convex relaxation, finding global minimizers of the relaxed problem and subsequent projection to the original non-convex set. Compared to existing fusion techniques, we thus compute guaranteed silhouette-consistent reconstructions without constraining the search space and without extending the original stereo-weighted cost functional by heuristic ballooning terms or more sophisticated balancing terms. Compared to classical local fusion techniques, the proposed formulation does not depend on initialization and leads to a more tractable numerical scheme by removing the bias near the visual hull boundary. In experiments on several challenging real data sets we show the advantages of silhouette-consistency in the reconstruction of small-scale structures which cannot be restored by state-of-the-art stereo algorithms.

In the next section, we will briefly review the formulation of stereo-based multiview reconstruction as a weighted minimal surface problem. In Section 3 we will show that the integration of stereo and silhouette constraints can be formulated as a problem of minimizing a convex functional over the convex set of silhouette-consistent functions. In Section 4 we provide details on the numerical implementation of the constrained convex optimization. In Section 5, we show experimental results on several real data sets which demonstrate the advantages of silhouette consistency for the reconstruction of fine-scale structures. We conclude with a brief summary.

2 3D Reconstruction as a Minimal Surface Problem

Let $V \subset \mathbb{R}^3$ be a volume, which contains the scene of interest, and $I_1, \ldots, I_n : \Omega \to \mathbb{R}^3$ a collection of calibrated color images with perspective projections $\pi_1, \ldots, \pi_n$. Let $S_1, \ldots, S_n \subset \Omega$ be the observed projections of the 3D object and $\rho : V \to [0, 1]$ be a photoconsistency map measuring the discrepancy among various image projections. In particular, low values of $\rho(x)$ indicate a strong agreement from different cameras on the observed image patches, indicating a high likelihood that the surface passes through the given point. More details on the computation of photoconsistency will be given in Section 4.1.

With the above definitions, multiview reconstruction can be done by minimizing the classical energy [8]:

$$E(S) = \int_S \rho(x)\, dS. \tag{1}$$

The reconstruction is therefore given by a minimal surface measured in a Riemannian metric that favors boundaries along photoconsistent locations. While

local optimization techniques (using coarse-to-fine strategies) provide useful reconstructions, there is little guarantee regarding optimality of the solutions. In fact, the question of optimality is somewhat meaningless, as the global minimum of (1) is obviously the empty set. A remedy to this problem is to either constrain the search space around the visual hull [12], or to add regional balancing terms to the cost functional using balloon forces [10] or heuristically constructed regional terms [21,22,23]. Global optima of respective cost functionals can then be computed either in a spatially discrete setting using graph cuts [12,10,23] or in a spatially continuous setting using convex relaxation techniques [22].

Nevertheless, two limitations of such methods are that firstly, the balancing regional terms are typically based on a number of somewhat heuristic assumptions and can often introduce a bias in the resulting segmentation. Secondly, the resulting reconstructions are not guaranteed to be silhouette-consistent in the sense that the projections of the surface do not necessarily coincide with the observed silhouettes.

3 Convex Integration of Silhouettes and Stereo

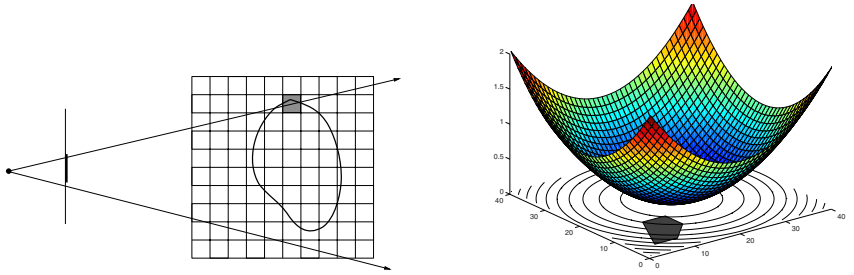
An alternative strategy to avoid trivial solution in the optimization of (1) is to impose silhouette alignment of the computed shape yielding the following constrained optimization problem:

$$\begin{aligned} E(S) &= \int_S \rho(x)\, dS, \\ \text{s. t.} \qquad & \pi_i(S) = S_i \quad \forall\, i = 1, \ldots, n. \end{aligned} \tag{2}$$

In order to cast (2) as a convex optimization problem, the surface S is represented implicitly by the characteristic function $u : V \to \{0, 1\}$ of the surface interior S_{int}. Hence, changes in the topology of S are handled automatically without reperametrization. With the implicit surface representation, we have the following constrained, convex energy functional equivalent to (2):

$$\begin{aligned} E(u) &= \int_V \rho(x) |\nabla u(x)|\, dx \\ &\text{s. t.} \qquad u \in \{0, 1\} \\ &\int_{R_{ij}} u(x)\, dR_{ij} \geq \delta \text{ if } j \in S_i \\ &\int_{R_{ij}} u(x)\, dR_{ij} = 0 \text{ if } j \notin S_i, \end{aligned} \tag{3}$$

where R_{ij} denotes the visual ray through pixel j of image i and $\delta > 0$ denotes the thickness, below which the given material becomes translucent. In the following we will set $\delta = 1$. Here we have rewritten the constraint in (2) in following form: For a silhouette-consistent shape *at least one* of the voxels along a visual ray



(a) (b)

Fig. 2. Silhouette constraints. (a) For a silhouette consistent shape at least one voxel along a visual ray through a silhouette pixel is occupied, whereas all voxels along a ray through a non-silhouette pixel are empty. The bold area on the image plane indicates the outlines of the observed object and the shaded voxel is an occupied one along the given viewing ray. (b) Multiview stereovision can be formulated as a convex optimization problem, where silhouette constraints determine a convex domain (shaded area) of admissible functions. Hence, global minimization is possible by using classical techniques like gradient descent.

through a silhouette pixel should be occupied, whereas *all* voxels along a ray determined by a non-silhouette pixel should be empty (see Fig. 2(a)).[1]

Due to the constraint that u is a binary-valued function, the minimization problem (3) is non-convex (because the space of binary functions is non-convex). By relaxing the binary constraint and allowing the function u to take on values in the interval $[0, 1]$, the optimization problem becomes that of minimizing a convex functional over a bounded convex set (see Fig. 2(b)):

$$\min_{u \in D} \int_V \rho(x) \, |\nabla u(x)| \, dx, \tag{4}$$

where

$$D := \left\{ u : V \to [0,1] \;\middle|\; \begin{array}{l} \int_{R_{ij}} u(x) \, dR_{ij} \geq 1 \;\; \text{if } j \in S_i \; \forall i, j \\ \int_{R_{ij}} u(x) \, dR_{ij} = 0 \;\; \text{if } j \notin S_i \; \forall i, j \end{array} \right\} \tag{5}$$

is the set of continuous valued functions u which fulfill the silhouette constraints for all images i and all rays j.

Proposition 1. *The set D of all silhouette-consistent functions defined in* (5) *forms a compact and convex set.*

Proof. D is obviously compact, since it is determined by multiple restricting inequalities.

In order to show the convexity, let $u_1, u_2 \in D$ be two elements of D. Then any convex combination $u = \alpha u_1 + (1 - \alpha) u_2$ with $\alpha \in [0, 1]$ is also an element in D. In particular, $u(x) \in [0, 1]$ for all x. Moreover,

[1] Note that in case of imperfect silhouettes the above constraints can be applied only to the areas of high confidence.

$$\int_{R_{ij}} u\, dR_{ij} = \alpha \int_{R_{ij}} u_1\, dR_{ij} + (1-\alpha) \int_{R_{ij}} u_2\, dR_{ij} \geq 1 \;\text{ if } j \in S_i,$$

and similarly

$$\int_{R_{ij}} u\, dR_{ij} = \alpha \int_{R_{ij}} u_1\, dR_{ij} + (1-\alpha) \int_{R_{ij}} u_2\, dR_{ij} = 0 \;\text{ if } j \notin S_i.$$

Thus $u \in D$. □

The above statement implies that a global minimum u^* of the relaxed problem (4) exists and can be computed, for example by a simple gradient descent procedure or by more efficient numerical schemes. A necessary condition for a minimum of (4) is stated by the associated Euler-Lagrange equation

$$0 = \operatorname{div}\left(\rho\, \frac{\nabla u}{|\nabla u|}\right) = \rho \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) + \langle \nabla \rho, \frac{\nabla u}{|\nabla u|} \rangle. \tag{6}$$

A numerical solution to this partial differential equation within the domain of admissible functions specified by the convex constraints in (4) will be detailed in Section 4.

Since we are interested in minimizers of the non-convex binary labeling problem (3), a straightforward methodology is to threshold the solution of the convex problem appropriately. Although this will not guarantee finding the global minimum of (3), the proposed strategy entails a series of advantages compared to classical local optimization techniques. Intuitively, extending the set of admissible functions, computing the global minimum over this domain and subsequently projecting to the nearest point within the original set is expected to give a more accurate estimate than a simple gradient descent procedure for smooth functionals. In particular, this approach always gives an upper bound for the energetic deviation of the computed solution from the global minimum.

Proposition 2. *Let u^* be a minimizer of (4) and let $D' \subset D$ be the set of binary silhouette-consistent functions. Let $u' \in D'$ be the (global) minimum of (3) and $\tilde{u}$ the solution obtained with the above procedure. Then, a bound $\gamma(u^\star, \tilde{u})$ exists such that*

$$E(\tilde{u}) - E(u') \leqq \gamma(u^\star, \tilde{u}).$$

Proof. The claim follows directly from the construction of the approach:

$$E(\tilde{u}) - E(u') \leqq E(\tilde{u}) - E(u^\star) =: \gamma(u^\star, \tilde{u}).$$

The inequality is due to $E(u^\star) \leqq E(u')$, since $D' \subset D$. □

The projection $\tilde{u} \in D'$ of a minimizer u^* onto D' can be computed by simple thresholding

$$\tilde{u}(x) = \begin{cases} 1, & \text{if } u^*(x) \geq \mu \\ 0, & \text{otherwise} \end{cases}, \tag{7}$$

where

$$\mu = \min \left\{ \left(\min_{i \in \{1,\ldots,n\}, j \in S_i} \max_{x \in R_{ij}} u^*(x) \right) \ , \ 0.5 \right\}. \tag{8}$$

This threshold μ provides the closest silhouette-consistent binary function to the solution of the relaxed problem.

Proposition 3. *The reconstructed surface exactly fulfills all silhouette constraints, i.e.* $\tilde{u} \in D'$.

Proof. Let R_{pq} be a given ray. For $q \notin S_p$ the silhouette constraint is fulfilled for any threshold $\mu \in (0, 1)$, since the labels $\tilde{u}(x)$ of all voxels x along the respective ray are 0. For $q \in S_p$, we have:

$$\mu \leq \min_{i \in \{1,\ldots,n\}, j \in S_i} \max_{x \in R_{ij}} u^*(x) \leq \max_{x \in R_{pq}} u^*(x).$$

This implies $\exists x \in R_{pq} : u^*(x) \geq \mu$ and hence $\exists x \in R_{pq} : \tilde{u}(x) = 1$. □

Thus, the proposed methodology has the following advantages:

- It allows to incorporate *exact* silhouette constraints without making premature hard decisions about voxel occupancy along each viewing ray passing through a silhouette pixel.
- It does not depend on initialization, since the relaxed functional is optimized globally.
- It leads to a simple and tractable numerical scheme, which does not rely on a locally estimated surface orientation, and thus does not introduce a bias near the visual hull boundary.

All these benefits will be investigated in the experimental section.

4 Implementation

This section will give more details on the implementation of the proposed approach.

4.1 Photoconsistency Estimation

In this paper, we propose a novel strategy for integration of silhouette and stereo information. The presented method operates on a precomputed photoconsistency map $\rho : V \to [0, 1]$ and is independent of its particular implementation. To validate its concept, we used the voting scheme proposed in [14] for photoconsistency computation. The choice of this technique was motivated by its robustness even without explicit visibility estimation and increased accuracy compared to traditional methods. See [14] for more details.

4.2 Constraint Realization

The minimization of (4) should be performed within the specified domain of admissible functions. To this end, one has to enforce the fulfillment of all constraints during the optimization process. A straightforward way to achieve this is to project the current estimate after each iteration to the next point in the restricted domain. For the first constraint in (4) this corresponds to just clipping values lying outside of the interval $[0, 1]$. The last constraint could be realized by starting with the visual hull as initialization (1 if the voxel is part of the visual hull and 0 otherwise) and keeping function values fixed outside of it. Since (4) is optimized globally, the initial guess has no impact on the result but only on the number of iterations needed until convergence. However, the second constraint in (4) requires more efforts. It states that the sum of the function values along each visual ray passing through a silhouette pixel should be at least 1. If this requirement is violated, the values of all voxels along the ray lying within the visual hull should be uniformly increased. Note that enforcing this constraint in a different order will generally produce a different result. However, when the evolution step is small enough, this issue is not crucial and can be ignored. In particular, the realization of this constraint can be done in parallel. In order to avoid computations of ray-volume intersections any time the constraint is checked, one can compute the set of relevant voxels to each viewing ray in a preprocessing step and store them in lists. However, the size of this data structure could grow significantly when the resolution of input images is high. For this reason, in our implementation we stored only the first and last voxel along each ray. Another important issue when using constraints is the frequency of enforcing them. In our implementation, we achieved a stable behavior when applying the first constraint after each optimization iteration and the silhouette constraints after each 10 iterations.

4.3 Linearization and Fixed-Point Iteration

In order to solve (6), we suggest to use a fixed point iteration scheme that transforms the nonlinear system into a sequence of linear systems. These can be efficiently solved with an iterative solver like successive over-relaxation (SOR).

The only source of nonlinearity in (6) is the diffusivity $g := \frac{\rho}{|\nabla u|}$. Starting with an initialization u^0, we can compute g and keep it constant. For constant g, (6) is linear and discretization yields a linear system of equations, which we solve with the SOR method. This means, we iteratively compute an update of u at voxel i by

$$u_i^{l,k+1} = (1-\omega)u_i^{l,k} + \omega \frac{\sum\limits_{j\in\mathcal{N}(i),j<i} g_{i\sim j}^l u_j^{l,k+1} + \sum\limits_{j\in\mathcal{N}(i),j>i} g_{i\sim j}^l u_j^{l,k}}{\sum\limits_{j\in\mathcal{N}(i)} g_{i\sim j}^l}, \tag{9}$$

where $\mathcal{N}(i)$ denotes the 6-neighborhood of i. Finally, $g_{i\sim j}$ denotes the diffusivity between voxel i and its neighbor j. It is defined as

$$g^l_{i\sim j} := \frac{g^l_i + g^l_j}{2}, \qquad g^l_i := \frac{\rho_i}{\sqrt{|\nabla u^l_i|^2 + \epsilon^2}}, \tag{10}$$

where $\epsilon := 0.001$ is a small constant that prevents the diffusivity to become infinite when $|\nabla u^l_i|^2 = 0$ and $|\nabla u^l_i|^2$ is approximated by standard central differences. The over-relaxation parameter ω has to be chosen in the interval $(0, 2)$ for the method to converge. The optimal value depends on the linear system to be solved. Empirically, for the specific problem at hand, we obtained a stable behavior for $\omega = 1.5$. After the linear solver yields a sufficiently good approximation (we iterated for $k = 1, ..., 10$), one can update the diffusivities and solve the next linear system. Iterations are stopped as soon as the energy decay in one iteration is in the area of number precision.

5 Experimental Results

We validate the proposed approach on a scene of a head statue with complex reflection properties containing thin structures (the pedestal); see Fig. 3[2]. Scenes of this type are a known challenge for variational stereo-based methods due to the violation of the Lambertian assumption and the presence of a regularizer, which introduces a bias towards shapes with small area. In particular, we implemented two classical paradigms in multiview stereovision: a weighted minimal surface formulation with a ballooning constraint [12] and a stereo propagating scheme [21,22]. The first method produces clear oversmoothing effects at concavities and small-scale structures. The second approach retrieves shape indentations but also leads to erroneous carving at thin parts and specularities. In contrast, the introduced technique produces accurate reconstructions of thin structures (the pedestal) as well as concave areas by incorporating silhouette constraints in the optimization process. Note that all three models are based on a classical minimal surface formulation but use different methodologies to avoid the empty surface as a solution. Note also that all three methods use silhouette information to restrict the ballooning, for initialization or to constrain the domain of admissible functions. Fig. 4 shows intermediate steps in the evolution process of the proposed approach. Although global minimization is used, which makes the choice of the initial guess irrelevant, we initialized with the visual hull to emphasize the basic difference of our method to classical local optimization procedures. Usually, local minimization techniques use the surface orientation to identify locations to deform the current shape in order to minimize the resulting reprojection error. However, this could lead to instabilities and introduce a bias near the visual hull boundary by involving surface points beyond the contour generator. In contrast, the introduced method recovers shape indentations effortlessly, while retaining silhouette alignment during the optimization process.

Fig. 5 shows a comparison between the proposed method and the approach of [19] on an image sequence of a statue of a Greek goddess. Note the visible

[2] The image sequences used in Figures 3 and 7 will be made available at http://www-cvpr.iai.uni-bonn.de/data/

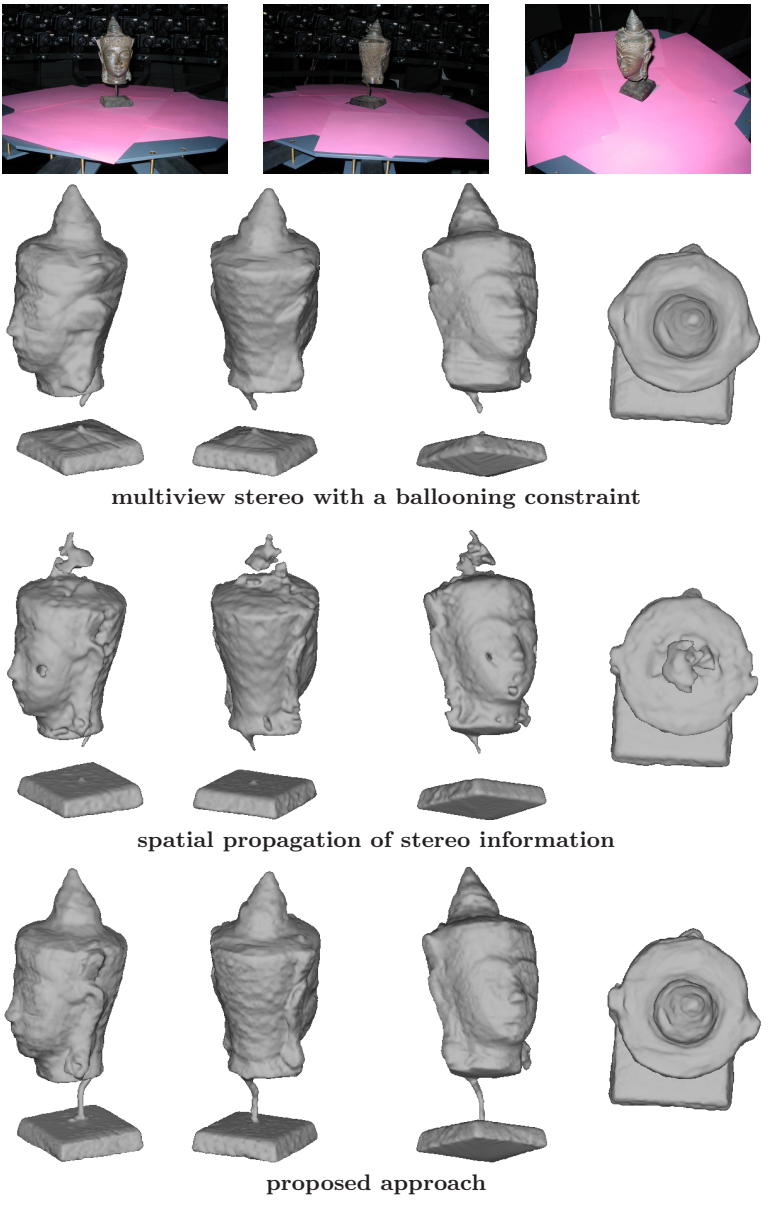


Fig. 3. Head sequence. *First row:* 3 of 33 input images of resolution 1024×768. *Second row:* Multiple views of the reconstruction with a model based on the combination of multiview stereo with a ballooning constraint [12]. *Third row:* Multiple views of the reconstruction obtained with a more elaborate model based on spatial propagation of stereo information [21,22]. *Fourth row:* Multiple views of the reconstruction with the proposed approach. While both methods fail to recover the pedestal of the statue due to oversmoothing effects or erroneous carving, the proposed approach recovers accurately all relevant details.

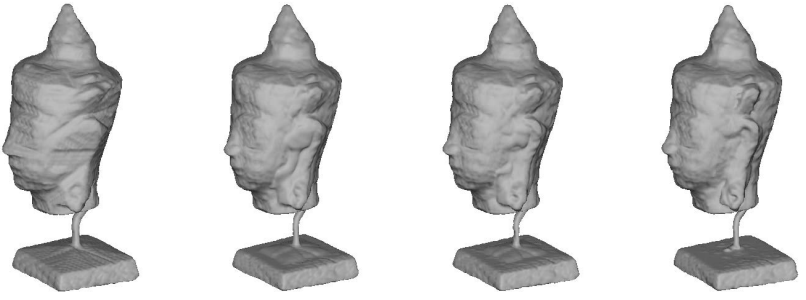


Fig. 4. Minimization process. Surface evolution starting from the visual hull, obtained by projecting the current estimate onto the original domain. Note that the presented method is able to generate accurate shapes starting from this initialization, since it does not take the local surface orientation into account.

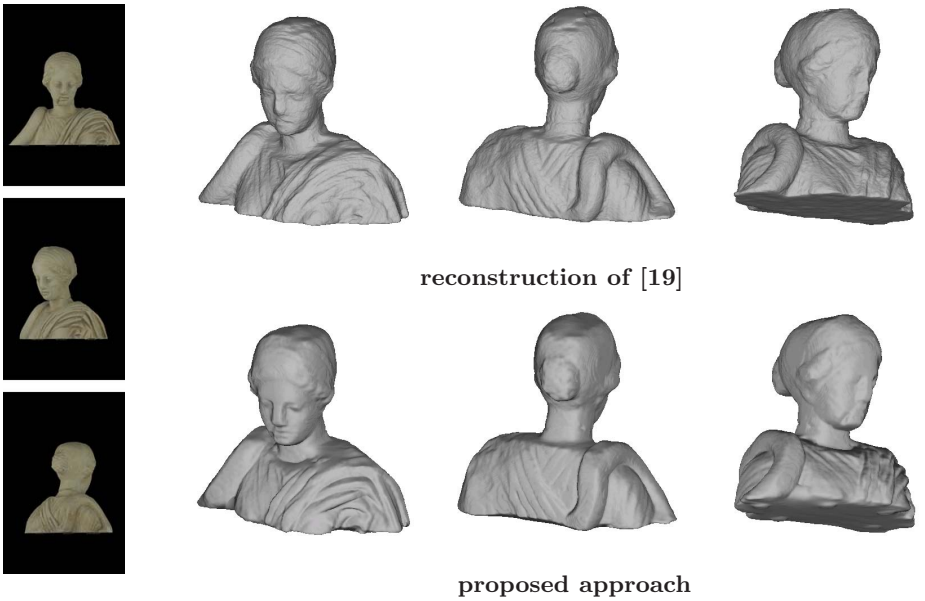


Fig. 5. Hygia sequence. 3 of 36 input images of resolution 2008×3040 and multiple views of the reconstructed surface compared to the reconstruction of [19]. Our result exhibits a higher grade of smoothness, while recovering surface details more accurately (for example the face and the creases of the cloth). Note that even the legs of the statue are reconstructed.

improvements of our reconstruction in the area of the face and the creases of the cloth. Note that even the legs of the statue are recovered. On the other hand, however, image noise is suppressed by increasing the grade of surface smoothness.

Fig. 6 and 7 illustrate two additional challenging image sequences. The first one is publicly available and captures a figurine of an ancient warrior. See

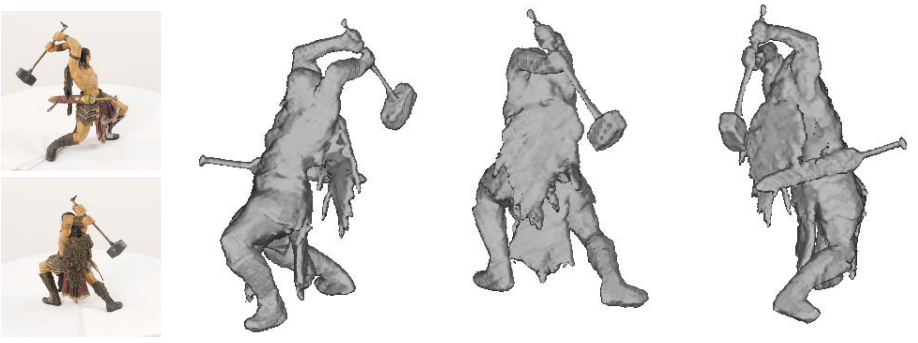


Fig. 6. Warrior sequence. 2 of 24 input images of resolution 1600×1600 and multiple views of the reconstructed surface. Note that thin structures (for example the handle of the hammer) as well as concavities (for example at the chest) are reconstructed accurately.

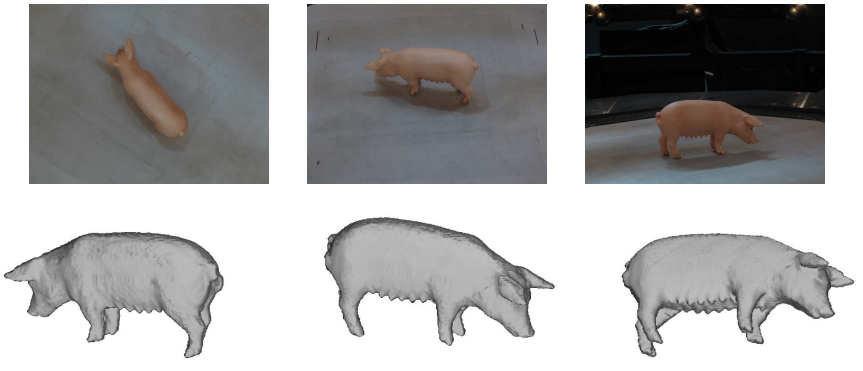


Fig. 7. Sow sequence. *First row:* 3 of 27 input images of resolution 1024×768. *Second row:* Multiple views of the reconstructed surface. Note the accurately reconstructed tits.

http://www-cvr.ai.uiuc.edu/ yfurukaw/research/mview/index.html for the data set including a reconstruction with the approach of [15]. Our result exhibits a high grade of smoothness, while preserving all fine geometric details. Similarly, the proposed approach generates a high-quality reconstruction of the sow figurine in Fig. 7 (see the accurately recovered tits). It is important to note that the absence of texture results in a minimal surface fulfilling silhouette consistency being generated, which allows to restore also homogeneous objects.

Apart from robustness, another crucial issue for a silhouette and stereo integration approach is the computational time needed. To this end, we used a GPU implementation of the presented method, where the SOR optimization in a red-black strategy as well as the imposed silhouette constraints run on the GPU. On a PC with 2.8 GHz and 4 GB of main memory, equipped with a NVIDIA

GeForce 8800 GTX graphics card, we measured computational times in the range of 30-60 seconds for all demonstrated experiments. Note that photoconsistency estimation is not included in these runtimes.

6 Conclusion

We proposed a novel framework for integrating silhouette and stereo information in 3D reconstruction from multiple images. The key idea is to cast multiview stereovision as a convex variational problem and to impose exact silhouette constraints by restricting the domain of feasible functions. Relaxation of the resulting formulation leads to the minimization of a convex functional over the convex set of silhouette-consistent functions, which can be performed in a globally optimal manner using classical techniques. A solution of the original problem is obtained by projecting the computed minimizer onto the corresponding restricted domain. In contrast to classical techniques for silhouette and stereo integration, it does leads to a more robust and tractable numerical scheme by avoiding hard decisions about voxel occupancy and removing the bias near the visual hull boundary. The proposed approach allows to compute accurate silhouette-consistent reconstructions for challenging real-world problems in less than one minute.

Acknowledgments

This research was supported by the German Research Foundation, grant # CR250/1-2. We thank Carlos Hernandez and Yasutaka Furukawa for providing the data sets in Figures 5 and 6. We thank Sudipta Sinha for sharing his results for Figure 5.

References

1. Baumgart, B.: Geometric modeling for computer vision. PhD thesis, Department of Computer Science, Stanford University, USA (1974)
2. Laurentini, A.: The visual hull concept for visual-based image understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence 16, 150–162 (1994)
3. Martin, W.N., Aggarwal, J.K.: Volumetric descriptions of objects from multiple views. IEEE Transactions on Pattern Analysis and Machine Intelligence 5, 150–158 (1983)
4. Cipolla, R., Giblin, P.: Visual motion of curves and surfaces. Cambridge University Press, Cambridge (2000)
5. Franco, J.S., Boyer, E.: Exact polyhedral visual hulls. In: Proceedings of the Fourteenth British Machine Vision Conference, Norwich, UK, pp. 329–338 (2003)
6. Yezzi, A., Soatto, S.: Stereoscopic segmentation. International Journal of Computer Vision 53, 31–43 (2003)
7. Seitz, S., Dyer, C.: Photorealistic scene reconstruction by voxel coloring. In: Proc. International Conference on Computer Vision and Pattern Recognition, pp. 1067–1073 (1997)

8. Faugeras, O., Keriven, R.: Variational principles, surface evolution, PDE's, level set methods, and the stereo problem. IEEE Transactions on Image Processing 7, 336–344 (1998)
9. Duan, Y., Yang, L., Qin, H., Samaras, D.: Shape reconstruction from 3D and 2D data using PDE-based deformable surfaces. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 238–251. Springer, Heidelberg (2004)
10. Lempitsky, V., Boykov, Y., Ivanov, D.: Oriented visibility for multiview reconstruction. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 226–238. Springer, Heidelberg (2006)
11. Matsumoto, Y., Fujimura, K., Kitamura, T.: Shape-from-silhouette/stereo and its application to 3D digitizer. In: Proceedings of Discrete Geometry for Computing Imagery, pp. 177–190 (1999)
12. Vogiatzis, G., Torr, P., Cippola, R.: Multi-view stereo via volumetric graph-cuts. In: Proc. International Conference on Computer Vision and Pattern Recognition, pp. 391–399 (2005)
13. Cross, G., Zisserman, A.: Surface reconstruction from multiple views using apparent contours and surface texture. In: Confluence of Computer Vision and Computer Graphics, Norwell, MA, USA, pp. 25–47. Kluwer Academic Publishers, Dordrecht (2000)
14. Esteban, C.H., Schmitt, F.: Silhouette and stereo fusion for 3D object modeling. Computer Vision and Image Understanding 96, 367–392 (2004)
15. Furukawa, Y., Ponce, J.: Carved visual hulls for image-based modeling. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 564–577. Springer, Heidelberg (2006)
16. Gargallo, P., Prados, E., Sturm, P.: Minimizing the reprojection error in surface reconstruction from images. In: Proceedings of the International Conference on Computer Vision, Janeiro, Brazil. IEEE Computer Society Press, Los Alamitos (2007)
17. Isidoro, J., Sclaroff, S.: Stochastic refinement of the visual hull to satisfy photometric and silhouette consistency constraints. In: Proc. International Conference on Computer Vision, Washington, DC, USA, pp. 1335–1342 (2003)
18. Tran, S., Davis, L.: 3D surface reconstruction using graph cuts with surface constraints. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 219–231. Springer, Heidelberg (2006)
19. Sinha, S., Mordohai, P., Pollefeys, M.: Multiview stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In: Proc. International Conference on Computer Vision, Rio de Janeiro, Brazil (2007)
20. Sinha, S., Pollefeys, M.: Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. In: Proc. International Conference on Computer Vision, Washington, DC, USA, pp. 349–356. IEEE Computer Society, Los Alamitos (2005)
21. Hernández, C., Vogiatzis, G., Cipolla, R.: Probabilistic visibility for multi-view stereo. In: Proc. International Conference on Computer Vision and Pattern Recognition, Minneapolis, Minnesota, USA. IEEE Computer Society, Los Alamitos (2007)
22. Kolev, K., Klodt, M., Brox, T., Cremers, D.: Propagated photoconsistency and convexity in variational multiview 3D reconstruction. In: Workshop on Photometric Analysis for Computer Vision, Rio de Janeiro, Brazil (2007)
23. Boykov, Y., Lempitsky, V.: From photohulls to photoflux optimization. In: Proc. British Machine Vision Conference, vol. 3, pp. 1149–1158 (2006)

Using Multiple Hypotheses to Improve Depth-Maps for Multi-View Stereo

Neill D.F. Campbell[1], George Vogiatzis[2],
Carlos Hernández[2], and Roberto Cipolla[1]

[1] Department of Engineering, University of Cambridge, Cambridge, UK
[2] Computer Vision Group, Toshiba Research Europe, Cambridge, UK

Abstract. We propose an algorithm to improve the quality of depth-maps used for Multi-View Stereo (MVS). Many existing MVS techniques make use of a two stage approach which estimates depth-maps from neighbouring images and then merges them to extract a final surface. Often the depth-maps used for the merging stage will contain outliers due to errors in the matching process. Traditional systems exploit redundancy in the image sequence (the surface is seen in many views), in order to make the final surface estimate robust to these outliers. In the case of sparse data sets there is often insufficient redundancy and thus performance degrades as the number of images decreases. In order to improve performance in these circumstances it is necessary to remove the outliers from the depth-maps. We identify the two main sources of outliers in a top performing algorithm: (1) spurious matches due to repeated texture and (2) matching failure due to occlusion, distortion and lack of texture. We propose two contributions to tackle these failure modes. Firstly, we store multiple depth hypotheses and use a spatial consistency constraint to extract the true depth. Secondly, we allow the algorithm to return an *unknown* state when the a true depth estimate cannot be found. By combining these in a discrete label MRF optimisation we are able to obtain high accuracy depth-maps with low numbers of outliers. We evaluate our algorithm in a multi-view stereo framework and find it to confer state-of-the-art performance with the leading techniques, in particular on the standard evaluation sparse data sets.

1 Introduction

The topic of multi-view stereo (MVS) reconstruction has become a growing area of interest in recent years with many differing techniques achieving a high degree of accuracy [1]. These techniques focus on producing watertight 3D models from a sequence of calibrated images of an object, where the intrinsic parameters and pose of the camera are known. In addition to providing a taxonomy of methods, [1] also provides a quantitative analysis of performance both in terms of accuracy and completeness. The top performers may be loosely divided into two groups. The first group make use of techniques such as correspondence estimation, local region growing and filtering to build up a final dense surface [13,15,16]. The

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 766–779, 2008.

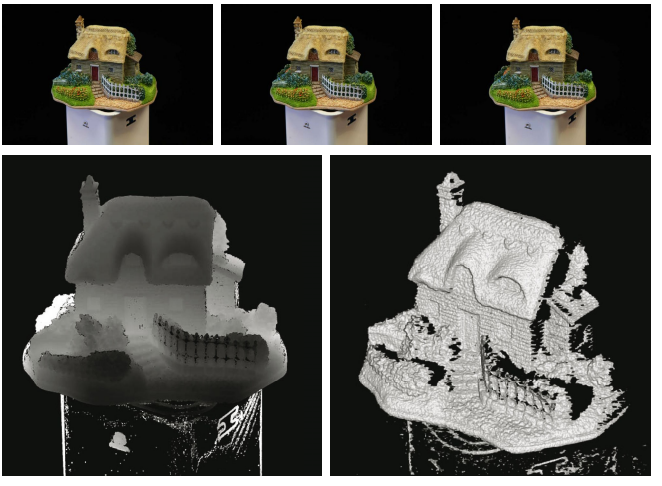


Fig. 1. Depth map obtained from only three images of a model house. The left image provides the recovered depth map which is rendered in the right image. As well as achieving a high degree of accuracy on surface detail our algorithm has correctly recovered the occlusion boundaries and removed outlying depth estimates.

second group make use of some form of global optimisation strategy on a volumetric representation to extract a surface [5,14,6,12,7]. A common strategy is to split the reconstruction process into two stages. The first is to estimate a series of depth-maps using local groups of the input images. The second stage then attempts to combine these into a global surface estimate, making use of registration and regularisation techniques. This two stage approach is an elegant formulation which allows different techniques to be chosen independently for the two stages. Some recent methods achieve a fast computation time by avoiding a global optimisation when merging depth-maps [17,18]. In this paper we focus on the first of the two stages — local depth-map estimation.

The estimation of local depth-maps is often performed using patch based methods [2]. The work of [5] proposed the use of Normalised Cross-Correlation (NCC) as the matching cost between two patches. This method offers good performance for textured objects and has been the basis of [7,6,19]. In the first stage of [5] a depth is estimated for each pixel independently. In the next stage the algorithm looks for consensus in depth estimates from multiple depth-maps. Since the individual depth-maps are known to contain outliers, this stage relies upon redundancy in the depth-maps to reject the them. In data-sets containing a large number of images (50-100) this approach performs quite well. In so called sparse data-sets (10-20 images) one expects very little redundancy in the reconstructed depth-maps, leading to a drop in reconstruction accuracy. This drop is actually observed in the performance of [5] in sparse data-sets with ground truth [1].

In this paper we show that if individual depth-maps are filtered for outliers prior to the fusion stage, good performance can be maintained in sparse data-sets. Our strategy is to collect a list of good hypotheses for the depth of each

pixel. We then chose the optimal depth for each pixel by enforcing consistency between neighbouring pixels in a depth-map. A crucial element of the filtering stage is the introduction of a possible *unknown* depth hypothesis for each pixel, which is selected by the algorithm when no consistent depth can be chosen. This pre-processing of the depth-maps allows the global fusion stage to operate on fewer outliers and consequently improve the performance under sparsity of data.

The rest of the paper is laid out as follows: In § 2 we review relevant prior work and discuss the differences of our approach. § 3 presents the use of NCC as a photo-consistency metric for estimating depth-maps and provides an overview of our algorithm to reduce outliers. § 4 provides the details of our depth-map estimation algorithm, in particular the optimisation process. In § 5 we show how to extend an existing MVS framework to include our depth-map estimation procedure for the purpose of the experimental evaluation provided in § 6. Here we display the improvements made to estimated depth-maps and also provided a quantitative evaluation of the MVS results. The paper concludes with our findings in § 7. This work was supported by a Schiff Scholarship and Toshiba Research Europe.

2 Previous Work

A taxonomy of the established methods for dense stereo may be found in [2]. Most of these methods use matching costs to assign each pixel to a set of disparity levels within the image. The earlier algorithms maintained relatively few separate levels and were more targeted towards depth based segmentation rather than detailed reconstruction. The latest algorithms [3] obtain depth-maps with greater accuracy. Since these algorithms only have pairs of images available, they can make no use of redundancy across multiple images in a data set and thus they use spatial regularisation and optimisation schemes which attempt to infer information about the depths. Whilst we also exploit a spatial regularisation constraint, we only allow the optimisation to choose from a set of discrete depths, well localised by the NCC peaks. This contrasts with methods which allow the depth of each pixel to vary continuously whilst minimising some cost function.

Some of the best performing algorithms make use of an occluded state. This may be via an explicit estimation of a disparity map, for example [20] or internally as part of an optimisation routine [4]. We make use of the unknown state in a similar manner however we also use it recognise the other failure modes of NCC matching, discussed in § 3, since they are indistinguishable.

The work of [5] proposed the robust NCC matching technique which we extend in our algorithm. Outlier rejection is accomplished through redundancy in the image sequence. The works of [7,6] have used derivatives of this technique with slight modifications, for example the inclusion of a Parzen window to filter the consensus matches in [6]. The work of [19] proposed a new, color normalised supersampling approach to correct for projective warping errors and also provided improved computation time with an efficient GPU implementation.

Recent work has demonstrated that depth-map estimation and integration paradigm may be used to produce accurate results with greatly reduced computation time [18] or real-time [17]. Again the reliance upon redundancy in the image sequence is paramount, for example the visibility computations of [17].

Since our contribution affects only the depth-map estimation, the global stage may be considered separately. The works of [23,24] present complementary algorithms for range image integration. Here, the depth-maps produced by our algorithm would provide a suitable set of range images. The use of a volumetric graph-cut to extract the surface was proposed in [14] and extended in [6] to include the robust NCC photoconsistency. Other works have shown the graph-cut formulation to perform well as a global optimisation stage [12,21].

The work of [22] uses multiple depth hypotheses as a result of reflections during the active 3D scanning of specular objects. Here a different framework, also based on spatial consistency, is used to reject false matches. The work of [26] makes use of multiple hypotheses for the related problem of new-view synthesis. They also make use of an MRF optimisation, here using a truncated quadratic kernel, to solve their synthesis problem.

3 Normalised Cross-Correlation for Photo-Consistency

Normalised Cross Correlation (NCC) may be used to define an error metric for matching two windows in different images. Figure 2 provides an example of using NCC and epipolar geometry to perform window based matching. If we fix a pixel location in a reference image, for each possible depth away from that pixel we get a corresponding pixel in the second image. By computing the NCC between windows centred in those two pixels we can define a matching score as a function of depth for the reference pixel. We refer to this function as the *correlation curve* of the pixel. A typical correlation curve will exhibit a very sharp peak at the correct depth, and possibly a number of secondary peaks in other depths.

In [5] a depth-map is generated for each input image using this matching technique for neighbouring images. For each pixel a number of correlation curves are computed (using a few of the neighbouring viewpoints) and the depth that gives rise to most peaks in those curves is selected as the depth for that pixel. See [5] or [6] for details. This process results in an independent depth estimate for each pixel. These depth estimates will unavoidably contain a significant percentage of outliers which must be dealt with in the subsequent step of [5] which is the volumetric fusion of multiple depth-maps. In data sets with a large number of images this is is overcome by the redundancy in the depth-estimates. The same surface point is expected to be covered by many different depth-maps, some of which will have the right depth estimate. In sparse data-sets however, each surface point may be seen by as few as two or three depth-maps. It is therefore crucial that outliers are minimised in the depth-map generation stage.

In this work we focus on the two most significant failure modes of NCC matching which are (1) the presence of repetitions in the texture and (2) complete matching failure due to occlusion, distortion and lack of texture. These are now described in more detail.

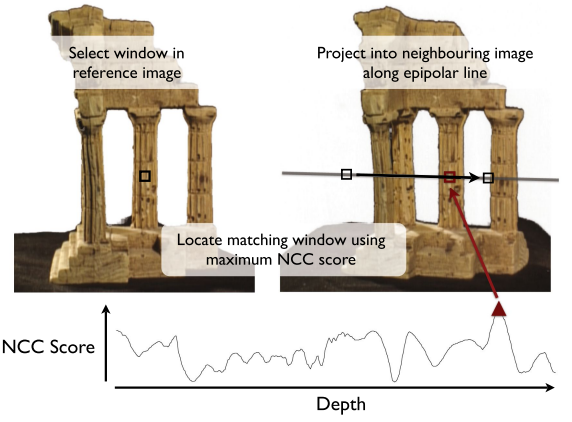


Fig. 2. Normalised Cross-Correlation based window matching

3.1 Repeating Texture

In general, there is no guarantee that the appearance of a patch is unique across the surface of the object. This results in correlation curve peaks at incorrect depths due to repeated texture — 'false' matches (Fig. 2). A larger window size is more likely to uniquely match to the true surface, reducing the number of false matches. However the associated peak will be broader and less well localised, reducing the accuracy of the depth estimate. The absolute value of the NCC score at a peak reflects how well the two windows match. Thus one might expect the peak with the maximum score to be the true peak. Unfortunately, the appearance of false matches due to repeated texture may result in false peaks having similar or even greater scores than the true surface peak (Fig. 3 (a)). To identify the correct peak, we propose to apply a spatial consistency constraint across neighbouring pixels in the depth-map. The underlying assumption is that if a peak corresponds to the true surface, the neighbouring pixels should have peaks at a similar depth. The exception to this is occlusion boundaries, which are however catered for under the next failure mode.

3.2 Matching Failure

The second failure mode is comprised of occlusion errors, distorted image windows (due to slanted surfaces) and lack of texture. In all of these cases, the correlation curve will not exhibit a peak at the true depth of the surface, resulting in only false peaks. Furthermore no spatial consistency can be enforced between the pixel in question and its neighbours. In this situation we would like to acknowledge that the depth at this pixel is unknown and should therefore offer no vote for the surface location.

In order to achieve these two goals we propose an optimisation strategy which makes use of a discrete label Markov Random Field (MRF). The MRF allows each pixel to choose a depth corresponding to one of the top NCC peaks which

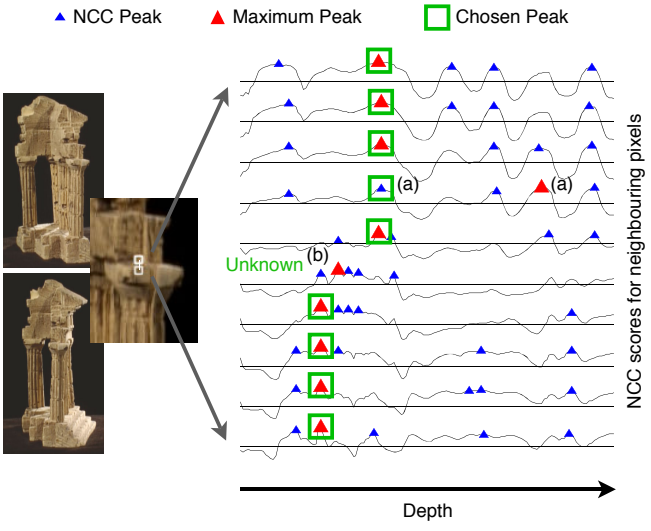


Fig. 3. Illustration of the MRF optimisation applied to neighbouring pixels. Existing method return the maximum peak which results in outliers in the depth estimate. The MRF optimisation corrects an outlier to the true surface peak (a) and introduces an unknown label at the occlusion boundary (b).

is spatially consistent with neighbouring pixels or select an *unknown* label to indicate that no such peak occurs and there is no correct depth estimate. This process means that the returned depth map should only contain accurate depths, estimated with a high degree of certainty, and an *unknown* label for pixels which have no certain associated depth. Figure 3 illustrates the optimisation for a 1D example of neighbouring pixels across an occlusion boundary.

4 Depth Map Estimation

Our proposed algorithm estimates the depth for each pixel in the input images. It proceeds in two stages: Initially we extract a set of possible depth values for each pixel using NCC as a matching metric. We then solve a multi-label discrete MRF model which yields the depth assignment for every pixel. One of the key features in this process is the inclusion of an *unknown* state in the MRF model. This state is selected when there is insufficient evidence for the correct depth to be found.

4.1 Candidate Depths

The input to our algorithm is a set of calibrated images $\mathcal{I}$ and the output is a set of corresponding depth-maps $\mathcal{D}$. In the following, we describe how to acquire a depth-map for a reference image $I_{\text{ref}} \in \mathcal{I}$. Let $N(I_{\text{ref}})$ denote a set of 'neighbouring' images to I_{ref}.

As proposed in § 3, we wish to obtain a hypothesis set of possible depths for each pixel $p_i \in I_{\text{ref}}$. Taking each pixel in turn, we project the epipolar ray into a second image $I_n \in I_{\text{ref}}$ and sample the NCC matching score over a depth range $\rho_i(z)$. We compute the score using a rectangular window centred at the projected image co-ordinates. One of the advantages of the multiple depth hypotheses is the ability to use a smaller matching window to provide a faster computation and improved localisation of the surface. Once we have obtained the sampled ray we store the top K peaks $\hat{\rho}_i(z_{i,k}), k \in [1, K]$ with the greatest NCC score for each pixel. Depending on the number of images available, and the width of the camera baseline, this process may be repeated for other neighbouring images. We then continue to the optimisation stage with a set of the best K possible depths, and their corresponding NCC scores, over all neighbouring images of I_{ref}.

4.2 MRF Formulation

At this stage a set of candidate depths $\hat{\rho}_i(z_{i,k}), k \in [1, K]$, for each pixel p_i in the reference image I_{ref} has been assigned and we wish to determine the correct depth map label for each pixel. As described in § 3, we also make use of an *unknown* state to account for the failure modes of NCC matching.

We model the problem as a discrete MRF where each each pixel has a set of up to $(K + 1)$ labels. The first K labels, fewer if an insufficient number of peaks were found during the matching stage, correspond to the peaks in the NCC function and have associated depths $z_{i,k} \in \mathcal{Z}_i$ and scores $\hat{\rho}_i(z_{i,k})$. The final state is the *unknown* state $\mathcal{U}$. If the optimisation returns this state, the pixel is not assigned a depth in the final depth map. For each pixel we therefore form an augmented label set $z'_{i,k} \in \{\mathcal{Z}_i, \mathcal{U}\}$ to include the unknown state.

The optimisation assigns a label $\bar{k}_i \in \{1 \ldots K, \mathcal{U})\}$ to each pixel p_i. The cost function to be minimised consists of unary potentials for each pixel and pairwise interactions over first order cliques. The cost of a labelling $\bar{\mathbf{k}} = \{\bar{k}_i\}$ is expressed as

$$E\left(\bar{\mathbf{k}}\right) = \sum_i \phi(\bar{k}_i) + \sum_{(i,j)} \psi(\bar{k}_i, \bar{k}_j) \tag{1}$$

where i denotes a pixel and (i, j) denote neighbouring pixels.

The following sections discuss the formulation of the unary potentials $\phi(\cdot)$ and pairwise interactions $\psi(\cdot, \cdot)$.

4.3 Unary Potentials

The unary labelling cost is derived from the NCC score of the peak. We wish to penalise peaks with a lower matching score since they are more likely to correspond to an incorrect match due to occlusion or noise. The NCC process will always return a score in the range $[-1, 1]$. As is common practice, [6], we take an inverse exponential function to map this score to a positive cost.

The unary cost for the *unknown* state is set to a constant value ϕ_{U}. This term serves two purposes. Firstly it acts as a cut-off threshold for peaks with poor NCC

scores which have no pairwise support (neighbouring peaks of similar depth). This mostly accounts for peaks which are weakly matched due to distortion or noise. Secondly it acts as a truncation on the depth disparity cost of the pairwise term. By assigning a low pairwise cost between peaks and the *unknown* state, the constant unary cost will effectively act as a threshold on the depth disparity to handle the case of an occlusion boundary. Thus the final unary term is given by

$$\phi\left(k_i = x\right) = \begin{cases} \lambda\, e^{-\beta\, \hat{\rho}_i(z_{i,x})} & x \in [1 \ldots K] \\ \phi_{\mathrm{U}} & x = \quad \mathcal{U} \end{cases} . \tag{2}$$

4.4 Pairwise Interactions

The pairwise labelling cost is derived from the disparity in depths of neighbouring peaks. As has been previously mentioned, this term is not intended to provide a strong regularisation of the depth map. Instead it is used to try and determine the correct peak, corresponding to the true surface location, out of the returned peaks. We observe that the correct peak may not have the maximum score. Therefore if there is strong agreement on depth between neighbouring peaks, we take this to be the true location of the surface.

When dealing with the depth disparity term we are really considering surface orientation; whether the surface normal is pointing towards or away from the camera. Under a perspective projection camera model it is therefore necessary to correct for the absolute depth of the peaks rather than simply taking the difference in depth. We perform this correction by dividing by the average depth of the two peaks. The resulting pairwise term is given by

$$\psi\left(k_i = x, k_j = y\right) = \begin{cases} 2\dfrac{\left|z_{i,x} - z_{j,y}\right|}{(z_{i,x} + z_{j,y})} & x \in [1 \ldots K] \quad y \in [1 \ldots K] \\ \psi_{\mathrm{U}} & x = \quad \mathcal{U} \quad\; y \in [1 \ldots K] \\ \psi_{\mathrm{U}} & x \in [1 \ldots K] \quad y = \quad \mathcal{U} \\ 0 & x = \quad \mathcal{U} \quad\; y = \quad \mathcal{U} \end{cases} . \tag{3}$$

We set ψ_{U} to a small value to encourage regions with many pixels labelled as *unknown* to coalesce. This acts as a further stage of noise reduction since it prevents spurious peaks with high scores but no surrounding support from appearing in regions of occlusion.

4.5 Optimisation

To obtain the final depth map we need to determine the optimal labelling $\hat{\mathbf{k}}$ such that

$$E(\,\hat{\mathbf{k}}\,) = \arg\min_{(\bar{\mathbf{k}})} \sum_i \phi(\bar{k}_i) + \sum_{(i,j)} \psi(\bar{k}_i, \bar{k}_j) \ . \tag{4}$$

Since in the general case this is an NP-hard problem we must use an approximate minimisation algorithm to achieve a solution. The most well-known techniques for solving problems of this nature are based on graph-cuts and belief propagation. Instead, we use the recently developed sequential tree-reweighted message passing algorithm, termed TRW-S, of [8]. This has been shown to outperform belief propagation and graph-cuts in tests on stereo matching using a discrete number of disparity levels. In addition to minimising the energy, the algorithm estimates a lower bound on the energy at each iteration which is useful in checking for convergence and evaluating the performance of the algorithm. We should note, however, that we are by no means guaranteed that the lower bound is attainable.

5 Extension to Multi-View Stereo Framework

As previously discussed, the detailed evaluation of [1] demonstrates that volumetric methods display state-of-the-art performance both in terms of accuracy and completeness. Some of the most successful create a 3D cost field within a volume and the reconstruction task is then to extract the optimal surface from this volume. Algorithms developed for segmentation problems are commonly used to extract the surface.

In order to evaluate the improvement to multi-view stereo we combined our depth map estimation with a modified implementation of the volumetric regularisation framework of [6]. This method uses a volumetric graph-cut to recover the surface from an array of voxels. Each voxel becomes a node in a 3D binary MRF where the voxel must be labelled as inside or outside the object. The MRF formulation allows for two terms in the cost function. The first is the unary foreground/background labelling cost. This encodes the likelihood that a particular voxel is part of the object or empty space. The recent work of [9] shows how depth maps may be used to evaluate a probabilistic visibility measure for each voxel in the volume. This term may be used to estimate whether or not the voxel in question resides in empty space and is therefore visible from the cameras. From this it is possible to derive an appropriate cost for the unary term related to the likelihood of visibility. The second term is the pairwise discontinuity cost. This term represents the likelihood that the surface boundary lies between two neighbouring voxels. This term may be derived directly from the individual depth maps projected into the volume.

In [25] the authors show that the energy cost is a discrete approximation to the sum of a weighted surface area of the boundary (the pairwise terms) and a weighted volume of the object (the unary terms). This framework is ideal for use with our depth maps since it provides global regularisation using all the available data. This is a key advantage of our approach. Rather than perform regularisation on individual depth maps to recover uncertain regions, we only return depths with a high degree of confidence associated with them. Thus other depth maps may be able to fill in the areas where a particular depth map is uncertain. In the event that there are still regions of the surface which are not

determined precisely by any of the depth maps, the regularisation should be performed by a global method which takes into account the data from all the depth maps rather than an amalgamation of estimates from individual depth maps.

5.1 Depth Map Acquisition

The first stage of the reconstruction process is to acquire the depths maps. Our method is to select an image and project rays into the nearest neighbouring images in a sequential process. We maintain a cumulative store of the K top scoring NCC peaks for each pixel. This provides an even greater degree of robustness against occlusion than the technique of [6] and is easier to implement in a parallel environment such as a GPU. Rather than requiring peaks from multiple images to fall in the same location, we only have to accurately observe a surface location in a single pair of images and rely on the surrounding support of peaks to identify the correct peaks. The speed of the depth map computation maybe increased by using the object silhouettes to avoid performing NCC matching calculations in regions outside the possible surface locations. Extraction of silhouettes for multi-view stereo may be performed as an automatic process [10].

5.2 Surface Recovery

Integrating our depth maps with the framework of [6] and [9] is a simple and elegant process. For the visibility volume we may project the same probability of visibility along each ray as [9] when we have a known depth. For pixels labelled as *unknown* we simply project a likelihood of 0.5 to indicate that this pixel provides no information about visibility. For the discontinuity cost we adopt a 'binning' approach. For each voxel in the discontinuity volume we take the sum of the projected depths of all the pixels in all the depth maps which fall inside the voxel, weighted by their NCC scores. If a pixel is labelled as *unkown* then it plays no part in the discontinuity cost. The final optimisation follows in the same manner as [6] with the graph-cut used to segment the volume. The iso-surface is extracted and smoothed using a snake [5] to perform 'intelligent' smoothing making use of the photoconsistency volume.

6 Experiments

6.1 Implementation

To improve the computation time for our depth maps we perform the NCC matching by taking advantage of the parallel processing and texture facilities of the GPU of modern graphics cards. The GPU code improves performance by up to an order of magnitude depending on the window size. On of the advantages of our method is the ability to use small windows which result in greater precision of the surface location but which introduce a significant amount of noise which will adversely affect many of the existing techniques. The use of the smaller window

also results in a greater saving in computational efficiency since the GPU offers improved performance with small kernels.

For the optimisation of the discrete MRF for the depth map we use the TRW-S implementation of Kolmogorov [8]. We also use Kolmogorov's implementation of the graph-cut algorithm [11]. Our implementation, running on a 3.0 GHz machine with an nVidia Quadro graphics card, can evaluate 900 NCC depth slices in 20 seconds for the temple sequence (image resolution 640×480). The TRW-S optimisation has a typical run time of 20 seconds for the same images. The final volumetric graph-cut typically runs in under 5 minutes for a 350^3 voxel array.

For all the experiments we used the following parameter values: $\beta = 1$, $\lambda = 1$, $\phi_{\mathrm{U}} = 0.04$ and $\psi_{\mathrm{U}} = 0.002$. We used an NCC window size of 5×5.

6.2 Depth Maps

Fig. 4 illustrates the improvement of our method over the voting schemes of [5,6]. Fig. 4 (b) shows the depth that would be determined by simply taking the NCC peak with the greatest score. Our method, implemented here with $K = 9$ peaks, is able to select the peak corresponding to true surface peak from the ranked candidate peaks and Fig. 4 (d) illustrates that a significant proportion of the true surface peaks are not the absolute maximum. We also observe that pixels are correctly labelled with the *unknown* state along occlusion boundaries and along areas such as the back wall of the temple and edges of the pillars where the surface normal is oriented away from the camera. Looking at the rendering of this depth-map and its neighbour, Fig. 4(e-g), we can observe that very few erroneous depths are recovered and we observe that the combination of the two depths maps align and complement each other rather than attempting to fill in the holes on the individual depth-maps which would impact the subsequent multi-view stereo global optimisation.

Fig. 5 shows the results on the 'cones' dataset which forms part of the standard dense stereo evaluations images and consists of a single stereo pair with the left image shown. Our depth-map again shows a high degree of detail on textured surfaces and we correctly identify occlusion boundaries with the *unknown* state. Further more the algorithm also correctly textures the failure modes of NCC by returning the *unknown* state in texture-less regions where the matching fails to accurately localise the surface.

6.3 Multi-View Stereo Evaluation

In order to evaluate the improvement of our depth-maps to multi-view stereo we ran our algorithm on the standard evaluation 'temple' dataset. The following table provides the accuracy and completeness measures of [1] against the ground-truth data for the object. In terms of both accuracy and completeness our results provide a significant improvement in both the sparse ring and ring datasets. In particular we observe that the results for the sparse ring offer greater accuracy

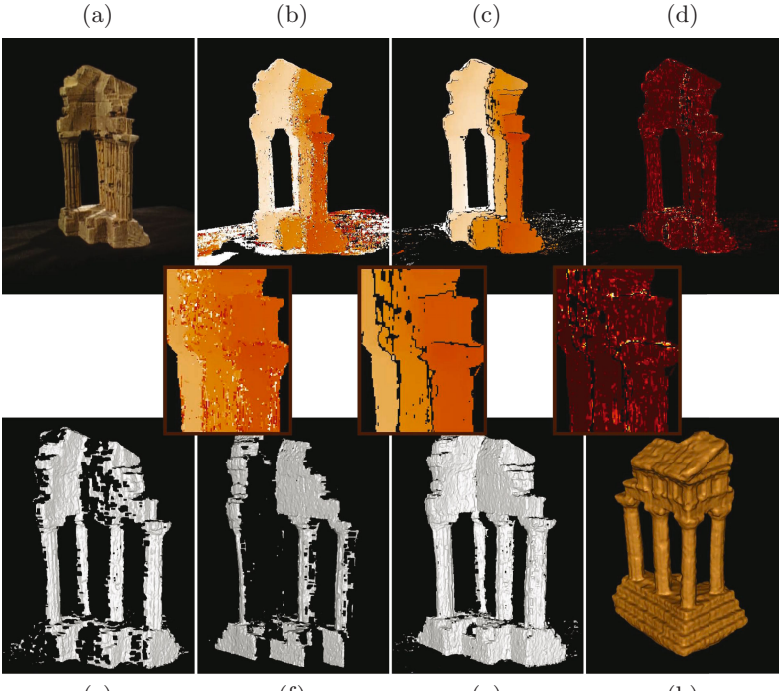


Fig. 4. Results of the depth map estimation algorithm. Two neighbouring images are combined with the reference image (a). If we simply took the NCC peak with the maximum score, as in [5], we would obtain (b). The result of our algorithm (c) shows a significant reduction in noise. We have corrected noisy estimates of the surface and the *unknown* state has also been used to clearly denote occlusion boundaries and remove poorly matched regions. The number of the correct surface peak returned, ranked by NCC score, is displayed in (d) where dark red indicates the peak with the greatest score. The rendered depth-map is shown in (e) along with the neighbouring depth-map (f) with (g) showing the two superimposed. The final reconstruction (h) for the sparse temple sequence (16 images) of [1].

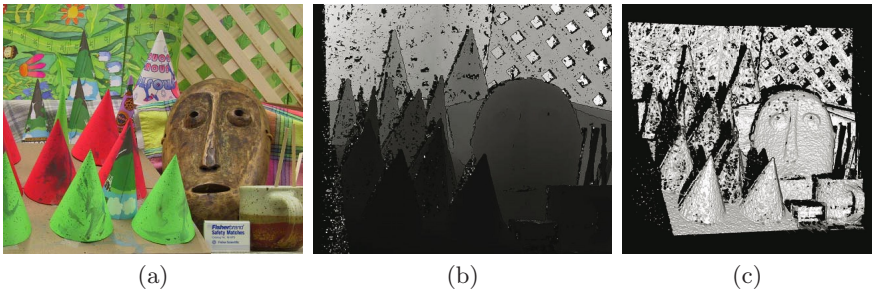


Fig. 5. Single view stereo results for the 'Cones' data set. The left image of the stereo pair is shown in (a) with the recovered depth-map in (b), rendered in (c).

than the other algorithms [3] running on the ring sequence (3 times as many images) with the exception of [5].

	Accuracy / Completeness		
	Full (312 images)	Ring (47 images)	SparseRing (16 images)
Our Results	**0.41mm / 99.9%**	**0.48mm / 99.4%**	**0.53mm / 98.6%**

7 Conclusions

The results of our experiments confirm that our method offers a significant improvement in performance over the current state-of-the-art reconstruction algorithms when running on sparse data sets. By explicitly accounting for the failure modes of the NCC matching technique we are able to produce depth-maps which accurately locate the true surface in noise, allowing the use of small matching windows. We are also able to identify when the surface estimate is inconsistent, due to lack of texture or occlusion, and label pixels as having unknown depths. Returning this unknown state, rather than providing a form of local regularisation, allows a subsequent global regularisation to be performed over all the depth-maps using the best possible data. If there are unknown surface regions which are not recovered by the depth-map a global regularisation scheme is in a much better position to estimate the surface since it has access to all of the depth-maps. This is particularly true in the case of the sparse ring temple dataset and we believe is primarily responsible for its improved performance over other methods. We also note that our depth-map estimation algorithm may be integrated with a variety of multi-view stereo algorithms [5,6,7,12,17,18,21] where is should confer similar increases in performance.

References

1. Seitz, S., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2006)
2. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Intl. Journal of Computer Vision 47(1–3) (2002)
3. `http://vision.middlebury.edu/`
4. Criminisi, A., Shotton, J., Blake, A., Rother, C., Torr, P.: Efficient dense stereo with occlusions for new view-synthesis by four-state dynamic programming. Intl. Journal of Computer Vision 71(1) (2007)
5. Hernández, C., Schmitt, F.: Silhouette and stereo fusion for 3d object modeling. Computer Vision and Image Understanding 96(3) (December 2004)
6. Vogiatzis, G., Hernández, C., Torr, P.H.S., Cipolla, R.: Multi-view stereo via volumetric graph-cuts and occlusion robust photo-consistency. IEEE Trans. Pattern Anal. Mach. Intell. 29(12) (2007)
7. Goesele, M., Curless, B., Seitz, S.: Multi-view stereo revisited. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2006)
8. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. IEEE Trans. Pattern Anal. Mach. Intell. 28(10) (2006)

9. Hernández, C., Vogiatzis, G., Cipolla, R.: Probabilistic visibility for multi-view stereo. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2007)
10. Campbell, N.D.F., Vogiatzis, G., Hernández, C., Cipolla, R.: Automatic 3d object segmentation in multiple views using volumetric graph-cuts. In: 18th British Machine Vision Conference, vol. 1 (2007)
11. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. Pattern Anal. Mach. Intell. 26(9) (September 2004)
12. Hornung, A., Kobbelt, L.: Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2006)
13. Furukawa, Y., Pons, J.: Accurate, dense, and robust multi-view stereopsis. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2007)
14. Vogiatzis, G., Torr, P., Cipolla, R.: Multi-view stereo via volumetric graph-cuts. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2005)
15. Habbecke, M., Kobbelt, L.: A surface-growing approach to multi-view stereo reconstruction. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2007)
16. Goesele, M., Snavely, N., Curless, B., Hoppe, H., Seitz, S.: Multi-view stereo for community photo collections. In: Proc. 11^{th} Intl. Conf. on Computer Vision (2007)
17. Merrell, P., Akbarzadeh, A., Wang, L., Mordohai, P., Frahm, J.-M., Yang, R., Nistér, D., Pollefeys, M.: Real-time visibility-based fusion of depth maps. In: Proc. 11^{th} Intl. Conf. on Computer Vision (2007)
18. Bradley, D., Boubekeur, T., Heidrich, W.: Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2008)
19. Hornung, A., Kobbelt, L.: Robust and efficient photoconsistency estimation for volumetric 3D reconstruction. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 179–190. Springer, Heidelberg (2006)
20. Sun, J., Li, Y., Kang, S.B., Shum, H.-Y.: Symmetric stereo matching for occlusion handling. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2005)
21. Sinha, S.N., Mordohai, P., Pollefeys, M.: Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In: Proc. 11^{th} Intl. Conf. on Computer Vision (2007)
22. Park, J., Kak, A.C.: Multi-peak range imaging for accurate 3D reconstruction of specular objects. In: Proc 6^{th} Asian Conf. on Computer Vision (2004)
23. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proc. of the ACM SIGGRAPH 1996 (1996)
24. Zach, C., Pock, T., Bischof, H.: A globally optimal algorithm for robust TV-L1 range image integration. In: Proc. 11^{th} Intl. Conf. on Computer Vision (2007)
25. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: Proc. 9^{th} Intl. Conf. on Computer Vision (2003)
26. Woodford, O.J., Reid, I.D., Fitzgibbon, A.W.: Efficient new view syntesis using pairwise dictionary priors. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2007)

Sparse Structures in L-Infinity Norm Minimization for Structure and Motion Reconstruction*

Yongduek Seo, Hyunjung Lee, and Sang Wook Lee

Department of Media Technology, Sogang University, Korea
{yndk,whitetobi,slee}@sogang.ac.kr

Abstract. This paper presents a study on how to numerically solve the feasibility test problem which is the core of the bisection algorithm for minimizing the L_∞ error functions. We consider a strategy that minimizes the maximum infeasibility. The minimization can be performed using several numerical computation methods, among which the barrier method and the primal-dual method are examined. In both of the methods, the inequalities are sequentially approximated by log-barrier functions. An initial feasible solution is found easily by the construction of the feasibility problem, and Newton-style update computes the optimal solution iteratively. When we apply the methods to the problem of estimating the structure and motion, every Newton update requires solving a very large system of linear equations. We show that the sparse bundle-adjustment technique, previously developed for structure and motion estimation, can be utilized during the Newton update. In the primal-dual interior-point method, in contrast to the barrier method, the sparse structure is all destroyed due to an extra constraint introduced for finding an initial solution. However, we show that this problem can be overcome by utilizing the matrix inversion lemma which allows us to exploit the sparsity in the same manner as in the barrier method. We finally show that the sparsity appears in both of the L_∞ formulations - linear programming and second-order cone programming.

1 Introduction

The method of L_∞ norm minimization has been shown to be very useful for geometric vision problems because it yields the global minimum rather than a local one [1,2,3]. Recent works exhibit the diversity of the L_∞ applications: one-dimensional cameras in robotics application[4], motion computation [5], dealing with outliers [6,7], increasing the computational efficiency [8,9], and tracking a deformable surface [10]. In addition, combined with the branch-and-bound technique, the L_∞ method can be used for solving more general problems involving the rotation parameters; Hartley and Kahl [11] solved the two view motion problem in the sense of global optimality.

An advantage of the L_∞ formulations is that since the objective function is convex, the choice of mumerical optimization method is not critical and any numerical method

* This work was supported by the Korea Science and Engineering Foundation(KOSEF) grant funded by the Korean government (MOST) (R01-2006-000-11374-0). This research is accomplished as the result of the research project for culture contents technology development supported by KOCCA.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 780–793, 2008.

Algorithm 1. Bisection method to minimize L_∞ norm

Input: initial upper(U)/lower(L) bounds, tolerance $\epsilon > 0$.

1: **repeat**
2: $\gamma := (L + U)/2$
3: Solve the feasibility problem (12)
4: **if** *feasible* **then** $U := \gamma$ **else** $L := \gamma$
5: **until** $U - L \leq \epsilon$

will result in the same global minimum in principle. Furthermore, the optimization itself can be done simply by adopting a general convex solver such as SeDuMi[12], CSDP[13], etc. The L_∞ method is gaining popularity not only because of the (quasi-) convexity but also because of its simplicity in implementation.

The bisection method shown in Algorithm 1 has been the main algorithm for minimizing the L_∞ error norm for various geometric vision problems [2,3]. We note that even though the feasibility test is the core problem to be solved to minimize the maximum residual error (γ in Algorithm 1), most of the previous approaches have resorted to general solvers like SeDuMi and have not paid much attention to the efficiency of the feasibility problem. As a matter of fact, the feasibility problem can be solved using various strategies such as minimizing the maximum infeasibility or minimizing the sum of infeasibilities [14,15]. Ke and Kanade [3] presented the formulation of minimizing the maximum infeasibility for the feasibility problem. Instead of the maximum (L_∞) re-projection error, they minimized the m-th largest re-projection error for robustness to outliers. This is basically a combinatorial problem; the solution was approximated by minimizing the sum of infeasibilities.

This paper focuses on the feasibility problem for the structure and motion (SAM) under the assumption of known rotation. Since the feasibility problem is the most time consuming and computationally demanding, an efficient algorithm for it will result in high speed computation of the L_∞ optimization.

We first present the formulation of the SAM problem and an explicit formulation for the feasibility problem. Among a few computational methods for the feasibility problem, we choose to use the method that minimizes the maximum infeasibility, and provide its mathematical formulation. Actual numerical minimization of the maximum infeasibility can be done with various algorithms. We consider two interior-point methods: the barrier method and the primal-dual interior-point method. They are all based on Newton updating schemes starting from an initial solution. We show that the Newton update in the barrier method has the same sparse structure as that of the bundle-adjustment [16,17]. This sparsity can be exploited in a similar manner as in the bundle adjustment to improve the speed of computation. We also show how to find an initial solution for the barrier method. The primal-dual method is known to outperform the barrier method for various problems [15]. To facilitate finding an initial solution, however, it introduces an additional linear inequality constraint which destroys the beneficial sparse structure. We show that the matrix inversion lemma is applicable and this allows us to take advantage of the sparsity. By using any of the two methods, we can avoid (linearized) algebraic computations that have been used for finding initial solutinos in

many geometric vision problems [16]. Therefore, we may simply start the optimization without worrying about the quality of the initial solution. This is another merit when we use the L_∞ optimization.

The rest of this paper is organized as follows. Section 2 presents the feasibility test problem for SAM and preliminary definitions. Section 3 shows the linear programming formulation and the sparse matrix appearing in the barrier method. Section 4 discusses the sparse structure for the case of the second-order cone programming. Section 5 shows how to employ the primal-dual algorithm instead of the barrier method without destroying the sparse structure. In Section 6, it is shown that the method of minimizing the maximum infeasibility can be utilized to accelerate the bisection algorithm as well. Finally, concluding remarks are given in Section 7.

2 SAM and the Feasibility Problem

Given an image measurement $[u_{ki1}, u_{ki2}]^\top$ of the i-th point $\mathbf{X}_i$ in 3D through the k-th camera $\mathtt{P}_k = [\mathtt{R}_k|\mathbf{t}_k]$, the residual vector is defined by

$$\mathbf{e}_{ik} = \left[u_{ik1} - \frac{\mathbf{r}_{k1}^\top \mathbf{X}_i + t_{k1}}{\mathbf{r}_{k3}^\top \mathbf{X}_i + t_{k3}} \,,\; u_{ik2} - \frac{\mathbf{r}_{k2}^\top \mathbf{X}_i + t_{k2}}{\mathbf{r}_{k3}^\top \mathbf{X}_i + t_{k3}} \right]^\top , \tag{1}$$

where $\mathbf{r}_{kn}^\top$ is the n-th row vector of $\mathtt{R}_k$, and t_{kn} the n-th component of the vector $\mathbf{t}_k$. The numbers of 3D points and cameras are N and K, respectively. The total number of image measurements is denoted by M ($M \leq NK$), and the index set I_M represents the set of the (i,k) pairs ($|I_M| = M$). Our goal is to compute the optimum for all the parameters of $\mathbf{X}_i$ and $\mathbf{t}_k$ assuming that the rotation matrices $\mathtt{R}_k$ are known *a priori*.

Before we precede further, the gauge (coordinate system) must be chosen. We select the same gauge as that in Kahl [2]; the 3D point $\mathbf{X}_N$ is set to $[0,0,0]$, and the last component of the translation $t_{K3} = 1$, that is, $\mathbf{t}_K = [t_{K1}, t_{K2}, 1]^\top$. The number of total parameters is then $P = 3(N-1) + 3K - 1$. By gathering all the unknowns into a column vector

$$\theta = [\mathbf{X}_1, \ldots, \mathbf{X}_{N-1}, \mathbf{t}_1, \ldots, \mathbf{t}_{K-1}, t_{K1}, t_{K2}]^\top, \tag{2}$$

the residual vector $\mathbf{e}_{ik}$ can be written as

$$\mathbf{e}_{ik} = \left[\frac{\mathbf{a}_{ik1}^\top \theta + b_{ik1}}{\mathbf{c}_{ik}^\top \theta + d_{ik}}, \quad \frac{\mathbf{a}_{ik2}^\top \theta + b_{ik2}}{\mathbf{c}_{ik}^\top \theta + d_{ik}} \right]^\top , \tag{3}$$

where $\mathbf{a}_{ikn}$, b_{ikn}, $n = 1, 2$, $\mathbf{c}_{ik}$, and d_{ik} are all coefficient vectors and scalars which consist of u_{ikn} and the elements of $\mathtt{R}_k$. Note that $\mathbf{a}_{ikn}$ has only five non-zero elements:

$$\mathbf{a}_{ikn}^\top (3i-2:3i) = u_{ikn}\mathbf{r}_{k3}^\top - \mathbf{r}_{kn}^\top \tag{4}$$

$$\mathbf{a}_{ikn}^\top (3(N-1)+3k) = u_{ikn} \tag{5}$$

$$\mathbf{a}_{ik1}^\top (3(N-1)+3k-2) = \mathbf{a}_{ik2}^\top (3(N-1)+3k-1) = -1\,. \tag{6}$$

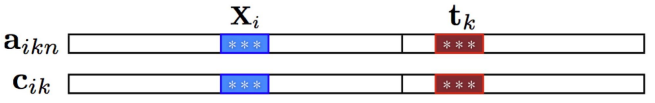


Fig. 1. Sparse shapes of $\mathbf{a}_{ikn}$ and $\mathbf{c}_{ik}$. The positions of non-zero elements are determined by the index i and k, respectively.

That is, $\mathbf{a}_{ikn}^{\top}(3i-2:3i)$ has three coefficients for $\mathbf{X}_i$, $\mathbf{a}_{ikn}^{\top}(3(N-1)+3k-n)$ has one for t_{kn}, and $\mathbf{a}_{ikn}^{\top}(3(N-1)+3k)$ has one for t_{k3}. Similarly, non-zero elements in the vector $\mathbf{c}_{ik}$ are:

$$\mathbf{c}_{ik}^{\top}(3i-2:3i) = \mathbf{r}_{k3}^{\top} \tag{7}$$

$$\mathbf{c}_{ik}^{\top}(3(N-1)+3k) = 1\ . \tag{8}$$

Figure 1 shows an illustration of the two vectors $\mathbf{a}_{ikn}$ and $\mathbf{c}_{ik}$. The L_2 norm $\|\mathbf{e}_{ik}\|_2$ is known to be a quasi-convex function. Its L_∞ norm $\|\mathbf{e}_{ik}\|_\infty$ is also a quasi-convex function because the absolute value (the L_1 norm) of each of the two functions

$$\left|\frac{\mathbf{a}_{ik1}^{\top}\theta + b_{ik1}}{\mathbf{c}_{ik}^{\top}\theta + d_{ik}}\right| \text{ and } \left|\frac{\mathbf{a}_{ik2}^{\top}\theta + b_{ik2}}{\mathbf{c}_{ik}^{\top}\theta + d_{ik}}\right| \tag{9}$$

is quasi-convex [9,11]. The L_1 norm $\|\mathbf{e}_{ik}\|_1$ is also quasi-convex as shown in [3]. We use $\|\cdot\|$ to represent any norm among the three norms: L_1, L_2, and L_∞.

Given a positive constant γ representing the maximum residual allowable, the solution space of θ is given by the intersection of all the constraints:

$$\|\mathbf{e}_{ik}(\theta)\| \le \gamma\ , \quad \forall ik \in I_M\ . \tag{10}$$

Therefore, the problem of minimizing the L_∞ error norm is given by

$$\begin{aligned} &\text{minimize } \ \gamma \\ &\text{subject to } \|\mathbf{e}_{ik}(\theta)\| \le \gamma,\ \forall ik \in I_M\ . \end{aligned} \tag{11}$$

The optimal solution can be found by the bisection method shown in Algorithm 1, in which the following feasibility test problem is solved repeatedly

$$\begin{aligned} &\text{find } \ \theta \\ &\text{s. t. } \ \|\mathtt{A}_{ik}\theta + \mathbf{b}_{ik}\| \le \gamma(\mathbf{c}_{ik}^{\top}\theta + d_{ik}), \forall ik \in I_M \end{aligned} \tag{12}$$

where $\mathtt{A}_{ik} = [\mathbf{a}_{ik1}|\mathbf{a}_{ik2}]^{\top} \in \mathbb{R}^{2\times P}$, $\mathbf{b}_i = [b_{ik1}, b_{ik2}]^{\top} \in \mathbb{R}^2$, and γ is a given constant during the bisectioning iteration.

2.1 The Feasibility Problem as a Minimization of Maximum Infeasibility

There are various computation methods for solving the feasibility test problem. The one we consider in this paper introduces an auxiliary variable s to find a strictly feasible solution of the inequalities or determine that none exists:

$$\begin{aligned} &\min\ s \\ &\text{s.t. } \|\mathtt{A}_{ik}\theta + \mathbf{b}_{ik}\| \le \gamma(\mathbf{c}_{ik}{}^{\top}\theta + d_{ik}) + s,\ \forall ik \in I_M\ . \end{aligned} \tag{13}$$

Note that the variable s represents a bound on the maximum infeasibility of the inequalities, and the goal of this problem is to drive s below zero. If $s \leq 0$ after the minimization, the problem is feasible, and a solution θ may be retrieved. Otherwise, $s > 0$ then the constraint set is infeasible, having an empty intersection.

We can re-write the problem in the standard form of convex optimization using augmented coefficient matrices. Let us construct the matrix $\tilde{\mathtt{A}}_{ik} = [\mathtt{A}_{ik}|\mathbf{0}]$ of size $2\times(P+1)$ by augmenting the matrix with a column of zeros, and the vectors $\tilde{\mathbf{c}} = [\gamma\mathbf{c}^\top, 1]^\top$ and $\tilde{\theta} = [\theta^\top, s]^\top$ of size $P+1$. Then we have

$$\begin{aligned} &\min \mathbf{f}^\top\tilde{\theta} \\ &\text{s.t. } \|\tilde{\mathtt{A}}_{ik}\tilde{\theta} + \mathbf{b}_{ik}\| \leq \tilde{\mathbf{c}}_{ik}^\top\tilde{\theta} + \gamma d_{ik}, \quad \forall ik \in I_M \\ &\quad \tilde{\mathbf{c}}_{ik}^\top\tilde{\theta} + d_{ik} \geq 0 \end{aligned} \tag{14}$$

where $\mathbf{f} = [\mathbf{0}, 1]^\top$ is a $P+1$ dimensional vector in which all components are zero except the last one. Note that (14) is a second-order cone programing problem if the L_2 norm function is adopted. It is a linear programming problem if L_∞ residual (9) is used.

3 Linear Programming

Using the L_∞ residual (9), we have the following linear programming problem:

$$\begin{aligned} &\min \mathbf{f}^\top\tilde{\theta} \\ &\text{s.t. } |\tilde{\mathbf{a}}_{ikn}^\top\tilde{\theta} + b_{ikn}| \leq \tilde{\mathbf{c}}_{ik}^\top\tilde{\theta} + \gamma d_{ik}\ , \quad n = 1, 2 \\ &\quad \tilde{\mathbf{c}}_{ik}^\top\tilde{\theta} + \gamma d_{ik} \geq 0 \end{aligned} \tag{15}$$

Note that this LP has $5M$ linear inequalities. The first constraint in (15) provides two linear constraints for each of $n = 1, 2$. The last inequality comes from the cheirality [18]. Thus, we have the following linear programming problem:

$$\begin{aligned} &\min \mathbf{f}^\top\tilde{\theta} \\ &\text{s.t} \\ &\quad (\tilde{\mathbf{a}}_{ik1} - \tilde{\mathbf{c}}_{ik})^\top\tilde{\theta} \leq \gamma d_{ik} - b_{ik1} \\ &\quad (-\tilde{\mathbf{a}}_{ik1} - \tilde{\mathbf{c}}_{ik})^\top\tilde{\theta} \leq \gamma d_{ik} + b_{ik1} \\ &\quad (\tilde{\mathbf{a}}_{ik2} - \tilde{\mathbf{c}}_{ik})^\top\tilde{\theta} \leq \gamma d_{ik} - b_{ik2} \\ &\quad (-\tilde{\mathbf{a}}_{ik2} - \tilde{\mathbf{c}}_{ik})^\top\tilde{\theta} \leq \gamma d_{ik} + b_{ik2} \\ &\quad -\tilde{\mathbf{c}}_{ik}^\top\tilde{\theta} \leq \gamma d_{ik}\ . \end{aligned} \tag{16}$$

Let us represent these five linear constraints by the matrix $\bar{\mathtt{A}}_{ik}$ of size $5 \times (P+1)$ and the 5-vector $\bar{\mathbf{b}}_{ik}$:

$$\bar{\mathtt{A}}_{ik} = \begin{bmatrix} (\tilde{\mathbf{a}}_{ik1} - \tilde{\mathbf{c}}_{ik1})^\top \\ (-\tilde{\mathbf{a}}_{ik1} - \tilde{\mathbf{c}}_{ik1})^\top \\ (\tilde{\mathbf{a}}_{ik2} - \tilde{\mathbf{c}}_{ik2})^\top \\ (-\tilde{\mathbf{a}}_{ik2} - \tilde{\mathbf{c}}_{ik2})^\top \\ -\tilde{\mathbf{c}}_{ik}^\top \end{bmatrix}, \quad \bar{\mathbf{b}}_{ik} = \begin{bmatrix} \gamma d_{ik} - b_{ik1} \\ \gamma d_{ik} + b_{ik1} \\ \gamma d_{ik} - b_{ik2} \\ \gamma d_{ik} + b_{ik2} \\ \gamma d_{ik} \end{bmatrix}\ . \tag{17}$$

Then the problem (16) can re-written as:

$$\begin{aligned} &\min \mathbf{f}^\top \tilde{\theta} \\ &\text{s.t } \bar{\mathtt{A}}_{ik}\tilde{\theta} \preceq \bar{\mathbf{b}}_{ik}\,, \quad \forall ik \in I_M, \end{aligned} \tag{18}$$

where $\preceq$ denotes the element-wise inequalities shown in (16).

We note that the coefficient vectors still keep the sparsity in spite of the vector additions and subtractions. For example, the vector $(\tilde{\mathbf{a}}_{ik1} - \tilde{\mathbf{c}}_{ik1})$ has seven non-zero elements including the last element (-1) newly introduced due to s. This is summarized as follows:

Result 1. *The matrix $\bar{\mathtt{A}}_{ik}$ has two 5×3 blocks, one for the structure $\mathbf{X}_i$ and the other for the motion $\mathbf{t}_k$, due to (4)-(8), and non-zero entries in the last column for the maximum infeasibility. In other words, it has non-zero columns at $3i-2:3i$, $3(N-1)+3k-2:3(N-1)+3k$, and $3(N-1)+3K$.*

3.1 The Barrier Method

For the time being, let us re-write briefly the LP (16) or equivalently (18) by using a single index m instead of ik:

$$\begin{aligned} &\min \mathbf{f}^\top \tilde{\theta} \\ &\text{s.t } \tilde{\mathbf{a}}_m^\top \tilde{\theta} \le b_m, \quad m \in I'_M \end{aligned} \tag{19}$$

where $\tilde{\mathbf{a}}_m$ is the vector of dimension $(P+1)$, and m is an index in the index set I'_M whose size is now five times larger than the original index set I_M.

Now we are ready to give a short description of the barrier method to find the optimal solution of this problem. The inequality constrained problem (19) can be approximated by an unconstrained problem as follows

$$\min\ E := \mathbf{f}^\top \tilde{\theta} - (1/t)\sum_m \log(b_m - \tilde{\mathbf{a}}_m^\top \tilde{\theta}) \tag{20}$$

The parameter $t > 0$ sets the accuracy of the approximation. As t increases, the quality of the approximation improves. The additional function

$$\phi(\theta) = -\sum_m \log(b_m - \tilde{\mathbf{a}}_m^\top \tilde{\theta}) \tag{21}$$

is called the *log barrier* for the problem (19). The domain is the set of points that satisfy the inequalities strictly:

$$\mathbf{dom}(\tilde{\theta}) = \{\tilde{\theta} | \tilde{\mathbf{a}}_m \tilde{\theta} < \tilde{\mathbf{c}}_m,\ \ m \in I'_M\} \tag{22}$$

When a strictly feasible starting point $\tilde{\theta}^{(0)}(t) \in \mathbf{dom}(\tilde{\theta})$ for a given t is provided, we can minimize the modified objective (20) through a Newton's method. An increment $\Delta\tilde{\theta}^{(l)}$ is computed based on a Taylor series expansion of E around $\tilde{\theta}^{(l)}$, and the solution is updated by $\tilde{\theta}^{(l+1)} = \tilde{\theta}^{(l)} + h\Delta\tilde{\theta}^{(l)}$; the step length h is determined by a search along the direction $\Delta\tilde{\theta}^{(l)}$. This process is repeated until we reach a convergence $\tilde{\theta}^*$ for the fixed t. Then, it is used as the starting point for the next unconstrained

Algorithm 2. Barrier method

Input: strictly feasible $\tilde{\theta} = \tilde{\theta}^{(0)}$, $t := t^{(0)} > 0$, tolerance $\epsilon > 0$.

1: **loop**
2: Centering step. Compute $\tilde{\theta}^*(t)$ by minimizing $t\mathbf{f}^\top\tilde{\theta} + \phi$, starting at $\tilde{\theta}$.
3: Update. $\tilde{\theta} := \tilde{\theta}^*(t)$.
4: Stopping criterion. **if** $5M/t < \epsilon$, **then stop**.
5: Increase t. $t := \mu t$. ($\mu > 0$).
6: **end loop**

minimization for an increased value of t. This outer iteration is repeated until $t \geq 5M/\epsilon$ so that we have a guaranteed ϵ-suboptimal solution of the original problem. A detailed theoretical explanation on this stopping criterion can be found in [15]. The barrier method is summarized in Algorithm 2.

3.2 Computing the Update Direction $\Delta\tilde{\theta}^{(l)}$

Specifically, the update direction is computed by solving the linear equation

$$\mathtt{H}\ \Delta\tilde{\theta}^{(l)} = -\mathbf{g} \tag{23}$$

where $\mathtt{H} = \nabla^2\phi(\tilde{\theta}^{(l)})$ is the hessian of the objective function and $\mathbf{g}(\tilde{\theta}^{(l)}) = t\mathbf{f} + \nabla\phi(\tilde{\theta}^{(l)})$ is the gradient. We note that $\mathtt{H}$ is given by

$$\mathtt{H} = \nabla^2\phi(\tilde{\theta}^{(l)}) = \sum_{ik} \bar{\mathtt{A}}_{ik}^\top \mathtt{W}_{ik} \bar{\mathtt{A}}_{ik} \tag{24}$$

where $\mathtt{W}_{ik}$ is a 5×5 diagonal matrix defined by $\bar{\mathtt{A}}_{ik}$ and $\bar{\mathbf{b}}_{ik}$ (which are given in (17)):

$$\mathtt{W}_{ik} = \mathrm{diag}\left(\bar{\mathbf{b}}_{ik} - \bar{\mathtt{A}}_{ik}\tilde{\theta}^{(l)}\right)^{-2}. \tag{25}$$

The hessian $\mathtt{H}$ has, along the diagonal, $(N-1)$ block-matrices of size 3×3 , $(K-1)$ block-matrices of 3×3 and one 2×2 block matrix. The last column and the last row are full of non-zero elements, which is a difference compared to the hessian of the bundle-adjustment in [17,16]. However, this does not affect the performance of the computation since $\Delta\tilde{\theta}^{(l)}$ is computed by applying the technique of the Schur complement; the $(N-1)$ blocks of 3×3 are inverted first to get a part of the solution and then $3K \times 3K$ matrix inversion is applied to get the other part of the solution.

Result 2. *The main part of the sparse structure of the matrix* $\mathtt{H}$ *is all the same as the one presented in [17,16] developed for the multiview bundle-adjustment algorithm.*

Figure 2 shows a sparseness of a data matrix $\mathtt{A}^\top$ from 24 image measurements from $N = 6$ points in $K = 4$ views. The plot was generated by our implementation in Matlab with a set of random data generation. One can notice the sparseness of the matrix from 120 ($= 5 \times 24$) linear constraints in total ($\mathbf{X}_6 = [0, 0, 0]$ and $t_{43} = 1$).

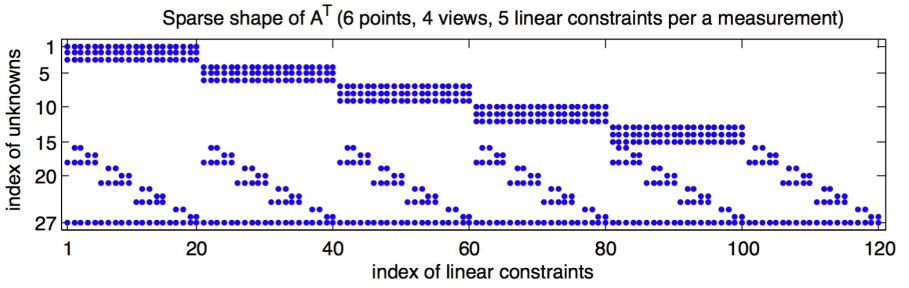


Fig. 2. Sparse structures of data matrix $\mathtt{A}^\top$ for $N = 6$ points and $K = 4$ views

3.3 Implementation with Schur Complement

The hessian $\mathtt{H}$ has the same sparse structure as mentioned. An example is given in Figure 3 for $N = 6$ points and $K = 4$ views. The total number of parameters is 27: one for the maximum infeasibility and 26 for the structure and motion ($= 3 * (6 - 1) + 3 * 4 - 1$). The red lines are to delimit three block sub-matrices.

Let $\mathtt{H} = \begin{bmatrix} \mathtt{U} & \mathtt{W} \\ \mathtt{W}^\top & \mathtt{V} \end{bmatrix}$. Now, it is clear that we can make use of the sparse structure for linear programming. Computing the inverse of the upper block matrix $\mathtt{U}$ is easy since it consists of 3×3 block diagonal matrices. Then, the rest of the sparse computation is based on solving a system of equations:

$$(\mathtt{V} - \mathtt{W}^\top \mathtt{U} \mathtt{W}) \mathbf{x}_\mathtt{V} = \mathbf{b}_\mathtt{V} \, . \tag{26}$$

One should note that the matrix $\mathtt{S} = \mathtt{V} - \mathtt{W}^\top \mathtt{U} \mathtt{W}$ for a large set of images has a banded structure because the matrix $\mathtt{W}$ becomes (band-diagonally) sparse when, e.g., the image

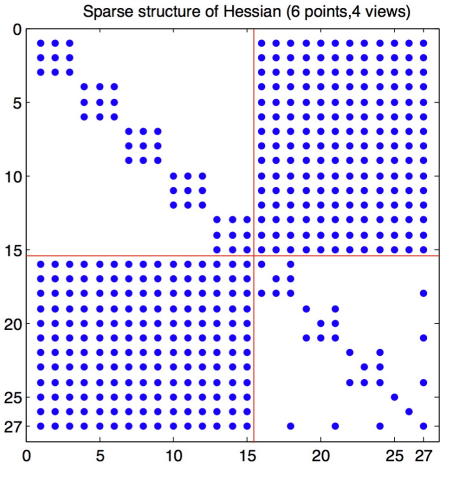


Fig. 3. Sparse structure of the hessian matrix $\mathtt{H}$ obtained from the data matrix shown in Figure 2 for $N = 6$ points and $K = 4$ views

tracks extend only over consecutive views. In this case an efficient sparse $\mathtt{LDL}^\top$ factorization is available, and it is essential to use the scheme of sparse $\mathtt{LDL}^\top$ factorization for an efficient computation, which is the case of SAM with many points from video cameras [16].

Result 3. *Our formulation adds only one more row and column in the hessian matrix compared to the hessian structure in [16]. Therefore, in addition to the sparseness of* $\mathtt{U}$*, we also have the same sparse structure for solving Equation (26).*

3.4 Initial Solution

An initial solution $\tilde{\theta}^{(0)} = [\theta^{(0)}; s^{(0)}]$ is necessary to start the Newton iteration in Algorithm 2. It can be found very easily:

$$\theta^{(0)} = \mathbf{0} \quad \text{and} \quad s^{(0)} = \max\{-b_m\} + c, \tag{27}$$

where $c > 0$ is a positive constant.

4 Second-Order Cone Programming

Applying the L_2 norm in (14), we come up with an SOCP. In this case, the barrier method adopts the *generalized logarithm* ψ of degree two for the log-barrier function

$$\psi(\mathbf{x}) = \log\left(x_3^2 - x_1^2 - x_2^2\right) \tag{28}$$

whose domain is defined by the second-order cone $\mathbf{dom}(\mathbf{x}) = \{\mathbf{x} \in \mathbb{R}^3 | \sqrt{x_1^2 + x_2^2} < x_3\}$. The barrier method minimizes the following unconstrained objective function while iteratively increasing t as before:

$$\min \quad E := t\mathbf{f}^\top\tilde{\theta} - \sum_{ik} \log(e_{ik3}^2 - e_{ik1}^2 - e_{ik2}^2) \tag{29}$$

where $\mathbf{e}_{ik} = [\tilde{\mathbf{a}}_{ik1}^\top\tilde{\theta} + b_{ik1}, \tilde{\mathbf{a}}_{ik2}^\top\tilde{\theta} + b_{ik2}, \tilde{\mathbf{c}}_{ik}^\top\tilde{\theta} + \gamma d_{ik}]^\top$. Now, let us define $\bar{\mathtt{A}}_{ik}$ of size $3 \times (P+1)$ as:

$$\bar{\mathtt{A}}_{ik} = [\tilde{\mathbf{a}}_{ik1} | \tilde{\mathbf{a}}_{ik2} | \tilde{\mathbf{c}}_{ik}]^\top . \tag{30}$$

The objective in (29) is then minimized by a Newton's method, solving the linear system of equations (23) where the hessian is now given by

$$\mathtt{H}_{\mathrm{SOCP}} = \sum_{ik} \bar{\mathtt{A}}_{ik}^\top \mathtt{H}_{ik} \bar{\mathtt{A}}_{ik} \tag{31}$$

where $\mathtt{H}_{ik} = \nabla^2\psi(\mathbf{e}_{ik})$ is of size 3×3. Examining the shape of the hessian matrix, we meet a sparse structure again:

Result 4. *The sparse structure of the hessian (31) is the same as the one in (24) for LP. So, it is the same as the one developed for the multiview bundle-adjustment in [17,16]. We have the additional weighting matrices* $\mathtt{W}_{ik}$ *and* $\mathtt{H}_{ik}$*, contrary to [17,16], but they do not change the sparse shape of the hessians* $\mathtt{H}$ *and* $\mathtt{H}_{\mathrm{SOCP}}$*, respectively.*

4.1 Initial Solution

A solution $\tilde{\theta}^{(0)}$ can be obtained: $\theta^{(0)} = \mathbf{0}$ and $s^{(0)} = \max\{\|\mathbf{b}_{ik}\|_2 - d_{ik}\} + c,\ c > 0$.

4.2 Stopping Criterion

From the theory of convex optimization over the second-order cone [15], the stopping criterion for this SOCP is $2M/t < \epsilon$ where M is the number of inequalities, 2 is the degree of the generalized logarithm ψ, and ϵ is the user-provided tolerance gap between the actual optimum and the true optimum. The 4-th line in Algorithm 2 should be changed accordingly.

5 Primal-Dual Interior-Point Method

Here we present a more efficient method for solving our SOCP which is the primal-dual interior-point method of Nesterov and Nemirovsky[19]. In general, primal-dual interior-point methods are often more efficient than the barrier method since they can exhibit better than linear convergence. It is reported that the primal-dual methods outperform the barrier method in various areas [15]. Our goal in this section is to show that the sparse structure can still be retained by using the matrix inversion lemma.

Here, we briefly introduce the primal-dual potential reduction method [19,20]. The method minimizes both of the primal and dual variables at the same time, contrary to the barrier method. The primal problem (14) is re-written below:

$$\begin{array}{l} \min \mathbf{f}^\top \tilde{\theta} \\ \text{s.t. } \|\tilde{\mathbf{A}}_{ik}\tilde{\theta} + \mathbf{b}_{ik}\|_2 \le \tilde{\mathbf{c}}_{ik}^\top \tilde{\theta} + \gamma d_{ik},\ \forall ik \in I_M \end{array} \tag{32}$$

where the linear inequality is omitted since it is superfluous. The dual problem of this is given by

$$\begin{array}{ll} \max & -\sum_{ik}(\mathbf{b}_{ik}^\top \mathbf{z}_{ik} + \gamma d_{ik} w_{ik}) \\ \text{s.t.} & \sum_{ik}(\tilde{\mathbf{A}}_{ik}^\top \mathbf{z}_i + \tilde{\mathbf{c}}_{ik} w_{ik}) = \mathbf{f}\ , \quad \forall ik \in I_M \\ & \|z_i\|_2 \le w_i \end{array} \tag{33}$$

The dual optimization variables are the vectors $\mathbf{z}_{ik} \in \mathbb{R}^2$ and $w_i \in \mathbb{R}$. Let $\lambda_{ik}^\top = [\mathbf{z}_{ik}^\top, w_{ik}]$, and $\mathbf{z}$ and $\mathbf{w}$ be the whole set of $\mathbf{z}_{ik}$'s and w_{ik}'s, respectively.

In the primal-dual potential reduction method [19], the potential function (34) below is minimized starting at an initial primal-dual solution $(\tilde{\theta}^{(0)}, \mathbf{z}^{(0)}, \mathbf{w}^{(0)})$:

$$\varphi(\tilde{\theta}, \mathbf{z}, \mathbf{w}) = (2M + \nu\sqrt{2M}) \log \eta + \sum_{ik} (\psi(\mathbf{e}_{ik}) + \psi(\lambda_{ik})) - 2N \log N \tag{34}$$

where $\nu \ge 1$ is an algorithm parameter, and η is the difference between the primal and dual objectives, called the duality gap:

$$\eta(\tilde{\theta}, \mathbf{z}, \mathbf{w}) = \mathbf{f}^\top \tilde{\theta} + \sum_{ik} (\mathbf{b}_{ik}^\top \mathbf{z}_{ik} + \gamma d_{ik} w_{ik})\ . \tag{35}$$

The theory asserts that if $\varphi \to -\infty$ then $\eta \to 0$ and $(\tilde{\theta}, \mathbf{z}, \mathbf{w})$ approaches optimality [19].

5.1 Initial Solutions

We already have shown how to find a primal solution $\tilde{\theta}^{(0)}$.To find a dual solutions, $\mathbf{z}^{(0)}$ and $\mathbf{w}^{(0)}$, an additional linear bound is included in the original problem without affecting the problem itself; this is called the big-M procedure. The dual of the modified problem is then such that a pair of dual solutions can always be computed. The modified problem is as follows:

$$\begin{array}{ll} \min & \mathbf{f}^\top \tilde{\theta} \\ \text{s.t.} & \|\tilde{\mathtt{A}}_{ik}\tilde{\theta} + \mathbf{b}_{ik}\|_2 \le \tilde{\mathbf{c}}_{ik}^\top \tilde{\theta} + \gamma d_{ik},\ \forall ik \in I_M \\ & \sum_{ik}(\tilde{\mathbf{c}}_{ik}\tilde{\theta} + d_{ik}) \le M_B \end{array} \tag{36}$$

and then its dual becomes

$$\begin{array}{ll} \max & -\sum_{ik}(\mathbf{b}_{ik}^\top \mathbf{z}_{ik} + d_{ik}(w_{ik} - \beta)) - \beta M_B \\ \text{s.t.} & \sum_{ik}(\tilde{\mathtt{A}}_{ik}^\top \mathbf{z}_i + \tilde{\mathbf{c}}_{ik}(w_{ik} - \beta)) = \mathbf{f}, \quad \forall ik \in I_M \\ & \|z_i\|_2 \le w_i, \\ & \beta \ge 0 \end{array} \tag{37}$$

Note that the problem is the same as the original one as long as the constant M_B is large enough. Practically, M_B is iteratively increased to keep the bound inactive during the optimization. Let us put $B = M_B - \sum_{ik}(\tilde{\mathbf{c}}_{ik}^\top \tilde{\theta} + d_{ik})$ and $\tilde{\mathbf{c}}_B = \sum_{ik} \tilde{\mathbf{c}}_{ik}$.

To find the initial dual solution, we first solve the linear constraint equation in (37) after setting $v_{ik} = w_{ik} - \beta$; this yields $\mathbf{z}_{ik}^{(0)}$ and v_{ik}. Since the system is under-determined, a least-norm solution can be used. Then, from the differences $\delta_{ik} = \|\mathbf{z}_{ik}^{(0)}\|_2 - v_{ik}$, we can find a strictly feasible solution

$$\beta^{(0)} = \max\{\max\{\delta_{ik}\}, 0\} + c, \quad c > 0\ , \tag{38}$$

and finally we have $w_{ik}^{(0)} = v_{ik} + \beta^{(0)}$.

5.2 Computing the Search Directions

In the primal-dual method, both of the primal and dual variables are updated via Newton iteration. The update $\Delta\tilde{\theta}$ is computed by solving a system of equations (exactly the same form as (23))

$$\mathtt{H}_{\mathrm{pd}}\ \Delta\tilde{\theta} = -\mathbf{g} \tag{39}$$

where $\bar{\mathtt{A}}_{ik}$ is defined in (30), the right hand side is given by

$$\mathbf{g} = \rho\mathbf{f} + \bar{\mathtt{A}}_{ik}^\top \nabla\psi(\mathbf{e}_{ik}) + \tilde{\mathbf{c}}_B/B \tag{40}$$

and the hessian $\mathtt{H}_{\mathrm{pd}}$ is given by

$$\mathtt{H}_{\mathrm{pd}} = \sum_{ik} \bar{\mathtt{A}}_{ik}^\top \mathtt{H}_{ik} \bar{\mathtt{A}}_{ik} + \frac{1}{B^2}\tilde{\mathbf{c}}_B\tilde{\mathbf{c}}_B^\top = \mathtt{H}_{\mathrm{SOCP}} + \frac{1}{B^2}\tilde{\mathbf{c}}_B\tilde{\mathbf{c}}_B^\top\ . \tag{41}$$

The dual direction $\Delta\lambda_{ik}$ is then computed using $\Delta\tilde{\theta}$:

$$\Delta\lambda_{ik} = -\rho\lambda_{ik} - \nabla\psi(\mathbf{e}_{ik}) - \nabla^2\psi(\mathbf{e}_{ik})\bar{\mathtt{A}}_{ik}\Delta\tilde{\theta} \tag{42}$$

which gives $\Delta\mathbf{z}_{ik}$ and Δw_{ik}. The outline of the method is given in Algorithm 3 [20].

Algorithm 3. Primal-dual method

Input: strictly feasible $(\tilde{\theta}, \mathbf{z}, \mathbf{w})$, tolerance $\epsilon > 0$.

1: **repeat**
2: Find primal and dual search directions by computing (41) and (42).
3: *Plane search.* Find $p, q \in \mathbb{R}$ that minimize $\psi(\tilde{\theta} + q\Delta\tilde{\theta}, \lambda_{ik} + q\Delta\lambda_{ik})$
4: *Update.* $\tilde{\theta} := \tilde{\theta} + p\Delta\tilde{\theta}$, $\lambda_i k := \lambda_{ik} + q\Delta\lambda_{ik}$.
5: **until** $\eta(\tilde{\theta}, \mathbf{z}, \mathbf{w}) \leq \epsilon$

5.3 Sparsity in Getting $\Delta\tilde{\theta}$

Note that the hessian $\mathtt{H}_{\mathrm{SOCP}}$ has the sparse structure but the last term $\tilde{\mathbf{c}}_B\tilde{\mathbf{c}}_B^\top$ has not in (41). A blind computation must deal with the symmetric positive definite but *non-sparse* matrix H_{pd} of size $(P+1) \times (P+1)$, which becomes unmanageable easily in our problem of multi-view SAM. Fortunately, exploiting the matrix inversion lemma allows us to utilize the sparse structure as before. The matrix inversion lemma for our case is as follows:

$$\left(\mathtt{H} + \frac{\tilde{\mathbf{c}}\tilde{\mathbf{c}}^\top}{B^2}\right)^{-1} = \mathtt{H}^{-1} - \mathtt{H}^{-1}\tilde{\mathbf{c}}\left(B^2 + \tilde{\mathbf{c}}^\top\mathtt{H}^{-1}\tilde{\mathbf{c}}\right)^{-1}\tilde{\mathbf{c}}^\top\mathtt{H}^{-1} \tag{43}$$

Therefore, we can solve two linear equations efficiently using the sparse structure of $\mathtt{H}_{\mathrm{SOCP}}$:

$$\mathtt{H}_{\mathrm{SOCP}}\mathbf{u} = \mathbf{g}\,, \qquad \mathtt{H}_{\mathrm{SOCP}}\mathbf{v} = \tilde{\mathbf{c}}_B \tag{44}$$

from which we have the solution of (39):

$$\Delta\tilde{\theta} = \mathbf{u} - \frac{\mathbf{v}\tilde{\mathbf{c}}_B^\top\mathbf{u}}{B^2 + \tilde{\mathbf{c}}_B^\top\mathbf{v}}\,. \tag{45}$$

Note that this final computation consists only of the vector inner products, a scalar division and some additive operations. In addition, solving the two equations (44) and (44) can be done simultaneously.

Result 5. *During the Newton update in the primal-dual method, the sparse structure still remains with the help of the matrix inversion lemma (43).*

6 Accelerating the Bisection Algorithm

During the iteration of minimizing the maximum infeasibility s, we may choose to stop the loop of the Newton iterations as soon as we obtain the feasibility ($s \leq 0$). Then, the bisection method (Algorithm 1) decreases the upper bound U and run the feasibility test again. This scheme usually results in a faster computation of the L_∞ optimization than continuing the Newton iterations.

On the other hand, we may continue the iterations even if the sign of s has changed from positive to negative. Based on our observations described below, we find this scheme useful in the early stage of the bisection method. Since s represents the maximum bound of the inequalities, the estimate θ^* corresponding to the maximum bound

will provide the maximum residual smaller than the upper bound γ in the bisection method. Therefore, we can choose $U = \min_{ik} \|\mathbf{e}_{ik}\|$ instead of $U = \gamma$ (the 4-th line in Algorithm 1). We notice that one of the fast methods shown in [2] is equivalent to this method.

We recommend using the method of continuing iterations even after the sign change of s only at the early stage of the bisection method when the gap $U - L$ is relatively large. For example, this scheme is very useful at the first bisection trial because, initially, the upper bound U needs to be large enough so that the bisection method may actually use a binary search. If the initial trial of the feasibility test is found to be infeasible, the upper bound must be increased. In this case, the bisection algorithm loses its binary searching capability but it is not easy to determine the increment. When the gap $U - L$ is small, on the other hand, the computational burden to decrease s tends to exceed the advantage we discuss above.

7 Conclusion

We investigated two interior-point methods to solve the feasibility test problem: the barrier method and the primal-dual potential reduction method. Basically, the interior-point methods are composed of the iterations of Newton update. Solving the problem of structure and motion with known rotation is shown to require dealing with a very large system of linear equations for the Newton update. In summary, we present findings that

- the sparse structure is very much similar to the one exploited in the bundle-adjustment,
- the technique that exploits sparsity is applicable to both of the L_∞ formulations, LP and SOCP.
- the formulation of the feasibility problem that minimizes the maximum infeasibility leads to the sparse structures,
- initial strictly feasible solutions to start Newton iteration can be obtained by problem constructions as shown,
- and the system of equations can be solved using *sparse* computation techniques appropriately developed for each of the problems.

For speeding up the bisection algorithm for the L_∞ optimization, it is recommended to use the improved low-level computation techniques we present in this paper as well as the high-level methods such as those presented in [2,9,21]. Recently, algorithms that search for rotation parameters have been presented [22,11] and a fast L_∞ optimizer will be more critical for those algorithms that involve searching over the rotation space. We are hopeful that our investigation shown in this paper expedites the development and use of such algorithms.

References

1. Hartley, R., Schaffalitzky, F.: L_∞ minimization in geometric reconstruction problems. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2004)
2. Kahl, F.: Multiple view geometry and the L_∞-norm. In: Proc. Int. Conf. on Computer Vision, Beijing, China, pp. 1002–1009 (2005)

3. Ke, Q., Kanade, T.: Quasiconvex optimization for robust geometric reconstruction. In: Proc. Int. Conf. on Computer Vision, Beijing, China (2005)
4. Aström, K., Enquist, O., Olsson, C., Kahl, F., Hartley, R.: An L-infinity approach to structure and motion problems for 1d-vision. In: ICCV 2007 (2007)
5. Sim, K., Hartley, R.: Recovering camera motion using L_∞ minimization. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2006)
6. Li, H.: A practical algorithm for L_∞ triangulation with outliers. In: IEEE Internatonal Conference on Computer Vision and Pattern Recognition (2007)
7. Sim, K., Hartley, R.: Removing outliers using the L_∞ norm. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2006)
8. Olsson, C., Anders, P., Eriksson, F.K.: Efficient optimization for L_∞ problems using pseudo-convexity. In: IEEE International Conference on Computer Vision (2007)
9. Seo, Y., Hartley, R.I.: A fast method to minimize L_∞ error norm for geometric vision problems. In: IEEE International Conference on Computer Vision (2007)
10. Salzmann, M., Hartley, R., Fua, P.: Convex optimization for deformable surface 3-d tracking. In: IEEE International Conference on Computer Vision (2007)
11. Hartley, R., Kahl, F.: Global optimization through searching rotation space and optimal estimation of the essential matrix. In: ICCV 2007 (2007)
12. Sturm, J.: Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. Optimization Methods and Software 11–12, 625–653 (1999)
13. Borchers, B.: CSDP, A C library for semidefinite programming. Optimization Methods and Software 11-12, 613–623 (1999)
14. Papadimitriou, C.H., Steiglitz, K.: Combinatorial optimization algorithms and complexity. Prentice Hall, Englewood Cliffs (1982)
15. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
16. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2004)
17. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment – a modern synthesis. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) ICCV-WS 1999. LNCS, vol. 1883. Springer, Heidelberg (2000)
18. Hartley, R.I.: The chirality. Int. Journal of Computer Vision 26, 41–61 (1998)
19. Nesterov, Y., Nemirovski, A.: Interior-point polynomial methods in convex programming. Studies in Applied Mathematics, vol. 13. SIAM, Philadelphia (1994)
20. Lobo, M., Vandenberghe, L., Boyd, S., Lebret, H.: Applications of second-order cone programming. Linear Algebra and its Applications 284, 193–228 (1998)
21. Agarwal, S., Snavely, N., Seitz, S.: Fast algorithms for l-infinity problems in multiview geometry. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2008)
22. Olsson, C., Kahl, F., Oskarsson, M.: Optimal estimation of perspective camera pose. In: International Conference on Pattern Recognition (2006)

Author Index