

Kostas Daniilidis  
Petros Maragos  
Nikos Paragios (Eds.)

LNCS 6315

# Computer Vision – ECCV 2010

11th European Conference on Computer Vision  
Heraklion, Crete, Greece, September 2010  
Proceedings, Part V

5  
Part V



 Springer

*Commenced Publication in 1973*

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

*Lancaster University, UK*

Takeo Kanade

*Carnegie Mellon University, Pittsburgh, PA, USA*

Josef Kittler

*University of Surrey, Guildford, UK*

Jon M. Kleinberg

*Cornell University, Ithaca, NY, USA*

Alfred Kobsa

*University of California, Irvine, CA, USA*

Friedemann Mattern

*ETH Zurich, Switzerland*

John C. Mitchell

*Stanford University, CA, USA*

Moni Naor

*Weizmann Institute of Science, Rehovot, Israel*

Oscar Nierstrasz

*University of Bern, Switzerland*

C. Pandu Rangan

*Indian Institute of Technology, Madras, India*

Bernhard Steffen

*TU Dortmund University, Germany*

Madhu Sudan

*Microsoft Research, Cambridge, MA, USA*

Demetri Terzopoulos

*University of California, Los Angeles, CA, USA*

Doug Tygar

*University of California, Berkeley, CA, USA*

Gerhard Weikum

*Max Planck Institute for Informatics, Saarbruecken, Germany*

Kostas Daniilidis Petros Maragos  
Nikos Paragios (Eds.)

# Computer Vision – ECCV 2010

11th European Conference on Computer Vision  
Heraklion, Crete, Greece, September 5-11, 2010  
Proceedings, Part V

## Volume Editors

Kostas Daniilidis  
GRASP Laboratory  
University of Pennsylvania  
3330 Walnut Street, Philadelphia, PA 19104, USA  
E-mail: kostas@cis.upenn.edu

Petros Maragos  
National Technical University of Athens  
School of Electrical and Computer Engineering  
15773 Athens, Greece  
E-mail: maragos@cs.ntua.gr

Nikos Paragios  
Ecole Centrale de Paris  
Department of Applied Mathematics  
Grande Voie des Vignes, 92295 Chatenay-Malabry, France  
E-mail: nikos.paragios@ecp.fr

Library of Congress Control Number: 2010933243

CR Subject Classification (1998): I.2.10, I.3, I.5, I.4, F.2.2, I.3.5

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

ISSN 0302-9743  
ISBN-10 3-642-15554-5 Springer Berlin Heidelberg New York  
ISBN-13 978-3-642-15554-3 Springer Berlin Heidelberg New York

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

springer.com

© Springer-Verlag Berlin Heidelberg 2010  
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India  
Printed on acid-free paper 06/3180

## Preface

The 2010 edition of the European Conference on Computer Vision was held in Heraklion, Crete. The call for papers attracted an absolute record of 1,174 submissions. We describe here the selection of the accepted papers:

- Thirty-eight area chairs were selected coming from Europe (18), USA and Canada (16), and Asia (4). Their selection was based on the following criteria: (1) Researchers who had served at least two times as Area Chairs within the past two years at major vision conferences were excluded; (2) Researchers who served as Area Chairs at the 2010 Computer Vision and Pattern Recognition were also excluded (exception: ECCV 2012 Program Chairs); (3) Minimization of overlap introduced by Area Chairs being former student and advisors; (4) 20% of the Area Chairs had never served before in a major conference; (5) The Area Chair selection process made all possible efforts to achieve a reasonable geographic distribution between countries, thematic areas and trends in computer vision.
- Each Area Chair was assigned by the Program Chairs between 28–32 papers. Based on paper content, the Area Chair recommended up to seven potential reviewers per paper. Such assignment was made using all reviewers in the database including the conflicting ones. The Program Chairs manually entered the missing conflict domains of approximately 300 reviewers. Based on the recommendation of the Area Chairs, three reviewers were selected per paper (with at least one being of the top three suggestions), with 99.7% being the recommendations of the Area Chairs. When this was not possible, senior reviewers were assigned to these papers by the Program Chairs, with the consent of the Area Chairs. Upon completion of this process there were 653 active reviewers in the system.
- Each reviewer got a maximum load of eight reviews—in a few cases we had nine papers when re-assignments were made manually because of hidden conflicts. Upon the completion of the reviews deadline, 38 reviews were missing. The Program Chairs proceeded with fast re-assignment of these papers to senior reviewers. Prior to the deadline of submitting the rebuttal by

the authors, all papers had three reviews. The distribution of the reviews was the following: 100 papers with an average score of weak accept and higher, 125 papers with an average score toward weak accept, 425 papers with an average score around borderline.

- For papers with strong consensus among reviewers, we introduced a procedure to handle potential overwriting of the recommendation by the Area Chair. In particular for all papers with weak accept and higher or with weak reject and lower, the Area Chair should have sought for an additional reviewer prior to the Area Chair meeting. The decision of the paper could have been changed if the additional reviewer was supporting the recommendation of the Area Chair, and the Area Chair was able to convince his/her group of Area Chairs of that decision.
- The discussion phase between the Area Chair and the reviewers was initiated once the review became available. The Area Chairs had to provide their identity to the reviewers. The discussion remained open until the Area Chair meeting that was held in Paris, June 5–6. Each Area Chair was paired to a buddy and the decisions for all papers were made jointly, or when needed using the opinion of other Area Chairs. The pairing was done considering conflicts, thematic proximity, and when possible geographic diversity. The Area Chairs were responsible for taking decisions on their papers. Prior to the Area Chair meeting, 92% of the consolidation reports and the decision suggestions had been made by the Area Chairs. These recommendations were used as a basis for the final decisions.
- Orals were discussed in groups of Area Chairs. Four groups were formed, with no direct conflict between paper conflicts and the participating Area Chairs. The Area Chair recommending a paper had to present the paper to the whole group and explain why such a contribution is worth being published as an oral. In most of the cases consensus was reached in the group, while in the cases where discrepancies existed between the Area Chairs' views, the decision was taken according to the majority of opinions.
- The final outcome of the Area Chair meeting, was 38 papers accepted for an oral presentation and 284 for poster. The percentage ratios of submissions/acceptance per area are the following:

Thematic area	# submitted	% over submitted	# accepted	% over accepted	% acceptance in area
Object and Scene Recognition	192	16.4%	66	20.3%	34.4%
Segmentation and Grouping	129	11.0%	28	8.6%	21.7%
Face, Gesture, Biometrics	125	10.6%	32	9.8%	25.6%
Motion and Tracking	119	10.1%	27	8.3%	22.7%
Statistical Models and Visual Learning	101	8.6%	30	9.2%	29.7%
Matching, Registration, Alignment	90	7.7%	21	6.5%	23.3%
Computational Imaging	74	6.3%	24	7.4%	32.4%
Multi-view Geometry	67	5.7%	24	7.4%	35.8%
Image Features	66	5.6%	17	5.2%	25.8%
Video and Event Characterization	62	5.3%	14	4.3%	22.6%
Shape Representation and Recognition	48	4.1%	19	5.8%	39.6%
Stereo	38	3.2%	4	1.2%	10.5%
Reflectance, Illumination, Color	37	3.2%	14	4.3%	37.8%
Medical Image Analysis	26	2.2%	5	1.5%	19.2%

- We received 14 complaints/reconsideration requests. All of them were sent to the Area Chairs who handled the papers. Based on the reviewers' arguments and the reaction of the Area Chair, three papers were accepted—as posters—on top of the 322 at the Area Chair meeting, bringing the total number of accepted papers to 325 or **27.6%**. The selection rate for the 38 orals was **3.2%**. The acceptance rate for the papers submitted by the group of Area Chairs was 39%.
- Award nominations were proposed by the Area and Program Chairs based on the reviews and the consolidation report. An external award committee was formed comprising David Fleet, Luc Van Gool, Bernt Schiele, Alan Yuille, Ramin Zabih. Additional reviews were considered for the nominated papers and the decision on the paper awards was made by the award committee. We thank the Area Chairs, Reviewers, Award Committee Members, and the General Chairs for their hard work and we gratefully acknowledge Microsoft Research for accommodating the ECCV needs by generously providing the CMT Conference Management Toolkit. We hope you enjoy the proceedings.

# Organization

## General Chairs

Argyros, Antonis	University of Crete/FORTH, Greece
Trahanias, Panos	University of Crete/FORTH, Greece
Tziritas, George	University of Crete, Greece

## Program Chairs

Daniilidis, Kostas	University of Pennsylvania, USA
Maragos, Petros	National Technical University of Athens, Greece
Paragios, Nikos	Ecole Centrale de Paris/INRIA Saclay île-de-France, France

## Workshops Chair

Kutulakos, Kyros	University of Toronto, Canada
------------------	-------------------------------

## Tutorials Chair

Lourakis, Manolis	FORTH, Greece
-------------------	---------------

## Demonstrations Chair

Kakadiaris, Ioannis	University of Houston, USA
---------------------	----------------------------

## Industrial Chair

Pavlidis, Ioannis	University of Houston, USA
-------------------	----------------------------

## Travel Grants Chair

Komodakis, Nikos	University of Crete, Greece
------------------	-----------------------------



## Area Chairs

Bach, Francis	INRIA Paris - Rocquencourt, France
Belongie, Serge	University of California-San Diego, USA
Bischof, Horst	Graz University of Technology, Austria
Black, Michael	Brown University, USA
Boyer, Edmond	INRIA Grenoble - Rhône-Alpes, France
Cootes, Tim	University of Manchester, UK
Dana, Kristin	Rutgers University, USA
Davis, Larry	University of Maryland, USA
Efros, Alyosha	Carnegie Mellon University, USA
Fermuller, Cornelia	University of Maryland, USA
Fitzgibbon, Andrew	Microsoft Research, Cambridge, UK
Jepson, Alan	University of Toronto, Canada
Kahl, Fredrik	Lund University, Sweden
Keriven, Renaud	Ecole des Ponts-ParisTech, France
Kimmel, Ron	Technion Institute of Technology, Ireland
Kolmogorov, Vladimir	University College of London, UK
Lepetit, Vincent	Ecole Polytechnique Federale de Lausanne, Switzerland
Matas, Jiri	Czech Technical University, Prague, Czech Republic
Metaxas, Dimitris	Rutgers University, USA
Navab, Nassir	Technical University of Munich, Germany
Nister, David	Microsoft Research, Redmont, USA
Perez, Patrick	THOMSON Research, France
Perona, Pietro	Caltech University, USA
Ramesh, Visvanathan	Siemens Corporate Research, USA
Raskar, Ramesh	Massachusetts Institute of Technology, USA
Samaras, Dimitris	State University of New York - Stony Brook, USA
Sato, Yoichi	University of Tokyo, Japan
Schmid, Cordelia	INRIA Grenoble - Rhône-Alpes, France
Schnoerr, Christoph	University of Heidelberg, Germany
Sebe, Nicu	University of Trento, Italy
Szeliski, Richard	Microsoft Research, Redmont, USA
Taskar, Ben	University of Pennsylvania, USA
Torr, Phil	Oxford Brookes University, UK
Torralba, Antonio	Massachusetts Institute of Technology, USA
Tuytelaars, Tinne	Katholieke Universiteit Leuven, Belgium
Weickert, Joachim	Saarland University, Germany
Weinshall, Daphna	Hebrew University of Jerusalem, Israel
Weiss, Yair	Hebrew University of Jerusalem, Israel

## 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
David Forsyth	University of Illinois, USA
Anders Heyden	Lund University, Sweden
Ales Leonardis	University of Ljubljana, Slovenia
Bernd Neumann	University of Hamburg, Germany
Mads Nielsen	IT University of Copenhagen, Denmark
Tomas Pajdla	CTU Prague, Czech Republic
Jean Ponce	Ecole Normale Superieure, France
Giulio Sandini	University of Genoa, Italy
Philip Torr	Oxford Brookes University, UK
David Vernon	Trinity College, Ireland
Andrew Zisserman	University of Oxford, UK

## Reviewers

Abd-Almageed, Wael	Bahlmann, Claus	Bougleux, Sebastien
Agapito, Lourdes	Baker, Simon	Boult, Terrance
Agarwal, Sameer	Ballan, Luca	Boureau, Y-Lan
Aggarwal, Gaurav	Barbu, Adrian	Bowden, Richard
Ahlberg, Juergen	Barnes, Nick	Boykov, Yuri
Ahonen, Timo	Barreto, Joao	Bradski, Gary
Ai, Haizhou	Bartlett, Marian	Bregler, Christoph
Alahari, Karteek	Bartoli, Adrien	Bremond, Francois
Aleman-Flores, Miguel	Batra, Dhruv	Bronstein, Alex
Aloimonos, Yiannis	Baust, Maximilian	Bronstein, Michael
Amberg, Brian	Beardsley, Paul	Brown, Matthew
Andreetto, Marco	Behera, Ardhendu	Brown, Michael
Angelopoulou, Elli	Beleznai, Csaba	Brox, Thomas
Ansar, Adnan	Ben-ezra, Moshe	Brubaker, Marcus
Arbel, Tal	Berg, Alexander	Bruckstein, Freddy
Arbelaez, Pablo	Berg, Tamara	Bruhn, Andres
Astroem, Kalle	Betke, Margrit	Buisson, Olivier
Athitsos, Vassilis	Bileschi, Stan	Burkhardt, Hans
August, Jonas	Birchfield, Stan	Burschka, Darius
Avraham, Tamar	Biswas, Soma	Caetano, Tiberio
Azzabou, Noura	Blanz, Volker	Cai, Deng
Babenko, Boris	Blaschko, Matthew	Calway, Andrew
Bagdanov, Andrew	Bobick, Aaron	Cappelli, Raffaele

Caputo, Barbara	Domke, Justin	Fua, Pascal
Carreira-Perpinan, Miguel	Donoser, Michael	Fuchs, Martin
Caselles, Vincent	Doretto, Gianfranco	Furukawa, Yasutaka
Cavallaro, Andrea	Douze, Matthijs	Fusiello, Andrea
Cham, Tat-Jen	Draper, Bruce	Gall, Juergen
Chandraker, Manmohan	Drbohlav, Ondrej	Gallagher, Andrew
Chandran, Sharat	Duan, Qi	Gao, Xiang
Chetverikov, Dmitry	Duchenne, Olivier	Gatica-Perez, Daniel
Chiu, Han-Pang	Duric, Zoran	Gee, James
Cho, Taeg Sang	Duygulu-Sahin, Pinar	Gehler, Peter
Chuang, Yung-Yu	Eklundh, Jan-Olof	Genc, Yakup
Chung, Albert C. S.	Elder, James	Georgescu, Bogdan
Chung, Moo	Elgammal, Ahmed	Geusebroek, Jan-Mark
Clark, James	Epshtein, Boris	Gevers, Theo
Cohen, Isaac	Eriksson, Anders	Geyer, Christopher
Collins, Robert	Espuny, Ferran	Ghosh, Abhijeet
Colombo, Carlo	Essa, Irfan	Glocker, Ben
Cord, Matthieu	Farhadi, Ali	Goecke, Roland
Corso, Jason	Farrell, Ryan	Goedeme, Toon
Costen, Nicholas	Favaro, Paolo	Goldberger, Jacob
Cour, Timothee	Fehr, Janis	Goldenstein, Siome
Crandall, David	Fei-Fei, Li	Goldluecke, Bastian
Cremers, Daniel	Felsberg, Michael	Gomes, Ryan
Criminisi, Antonio	Ferencz, Andras	Gong, Sean
Crowley, James	Fergus, Rob	Gorelick, Lena
Cui, Jinshi	Feris, Rogerio	Gould, Stephen
Cula, Oana	Ferrari, Vittorio	Grabner, Helmut
Dalalyan, Arnak	Ferryman, James	Grady, Leo
Darbon, Jerome	Fidler, Sanja	Grau, Oliver
Davis, James	Finlayson, Graham	Grauman, Kristen
Davison, Andrew	Fisher, Robert	Gross, Ralph
de Bruijne, Marleen	Flach, Boris	Grossmann, Etienne
De la Torre, Fernando	Fleet, David	Gruber, Amit
Dedeoglu, Goksel	Fletcher, Tom	Gulshan, Varun
Delong, Andrew	Florack, Luc	Guo, Guodong
Demirci, Stefanie	Flynn, Patrick	Gupta, Abhinav
Demirdjian, David	Foerstner, Wolfgang	Gupta, Mohit
Denzler, Joachim	Foroosh, Hassan	Habbecke, Martin
Deselaers, Thomas	Forssen, Per-Erik	Hager, Gregory
Dhome, Michel	Fowlkes, Charless	Hamid, Raffay
Dick, Anthony	Frahm, Jan-Michael	Han, Bohyung
Dickinson, Sven	Fraundorfer, Friedrich	Han, Tony
Divakaran, Ajay	Freeman, William	Hanbury, Allan
Dollar, Piotr	Frey, Brendan	Hancock, Edwin
	Fritz, Mario	Hasinoff, Samuel

Hassner, Tal	Kamarainen,	Larlus, Diane
Haussecker, Horst	Joni-Kristian	Latecki, Longin Jan
Hays, James	Kamberov, George	Lazebnik, Svetlana
He, Xuming	Kamberova, Gerda	Lee, ChanSu
Heas, Patrick	Kambhamettu, Chandra	Lee, Honglak
Hebert, Martial	Kanatani, Kenichi	Lee, Kyoung Mu
Heibel, T. Hauke	Kanaujia, Atul	Lee, Sang-Wook
Heidrich, Wolfgang	Kang, Sing Bing	Leibe, Bastian
Hernandez, Carlos	Kappes, Jörg	Leichter, Ido
Hilton, Adrian	Kavukcuoglu, Koray	Leistner, Christian
Hinterstoisser, Stefan	Kawakami, Rei	Lellmann, Jan
Hlavac, Vaclav	Ke, Qifa	Lempitsky, Victor
Hoiem, Derek	Kemelmacher, Ira	Lenzen, Frank
Hoogs, Anthony	Khamene, Ali	Leonardis, Ales
Hornegger, Joachim	Khan, Saad	Leung, Thomas
Hua, Gang	Kikinis, Ron	Levin, Anat
Huang, Rui	Kim, Seon Joo	Li, Chunming
Huang, Xiaolei	Kimia, Benjamin	Li, Gang
Huber, Daniel	Kittler, Josef	Li, Hongdong
Hudelot, Celine	Koch, Reinhard	Li, Hongsheng
Hussein, Mohamed	Koeser, Kevin	Li, Li-Jia
Huttenlocher, Dan	Kohli, Pushmeet	Li, Rui
Ihler, Alex	Kokiopoulou, Efi	Li, Ruonan
Ilic, Slobodan	Kokkinos, Iasonas	Li, Stan
Irschara, Arnold	Kolev, Kalin	Li, Yi
Ishikawa, Hiroshi	Komodakis, Nikos	Li, Yunpeng
Isler, Volkan	Konolige, Kurt	Liefeng, Bo
Jain, Prateek	Koschan, Andreas	Lim, Jongwoo
Jain, Viren	Kukelova, Zuzana	Lin, Stephen
Jamie Shotton, Jamie	Kulis, Brian	Lin, Zhe
Jegou, Herve	Kumar, M. Pawan	Ling, Haibin
Jenatton, Rodolphe	Kumar, Sanjiv	Little, Jim
Jermyn, Ian	Kuthirummal, Sujit	Liu, Ce
Ji, Hui	Kutulakos, Kyros	Liu, Jingen
Ji, Qiang	Kweon, In So	Liu, Qingshan
Jia, Jiaya	Ladicky, Lubor	Liu, Tyng-Luh
Jin, Hailin	Lai, Shang-Hong	Liu, Xiaoming
Jogan, Matjaz	Lalonde, Jean-Francois	Liu, Yanxi
Johnson, Micah	Lampert, Christoph	Liu, Yazhou
Joshi, Neel	Landon, George	Liu, Zicheng
Juan, Olivier	Langer, Michael	Lourakis, Manolis
Jurie, Frederic	Langs, Georg	Lovell, Brian
Kakadiaris, Ioannis	Lanman, Douglas	Lu, Le
Kale, Amit	Laptev, Ivan	Lucey, Simon

Luo, Jiebo	Mukaigawa, Yasuhiro	Peleg, Shmuel
Lyu, Siwei	Mulligan, Jane	Perera, A.G. Amitha
Ma, Xiaoxu	Munich, Mario	Perronnin, Florent
Mairal, Julien	Murino, Vittorio	Petrou, Maria
Maire, Michael	Namboodiri, Vinay	Petrovic, Vladimir
Maji, Subhransu	Narasimhan, Srinivasa	Peursum, Patrick
Maki, Atsuto	Narayanan, P.J.	Philbin, James
Makris, Dimitrios	Naroditsky, Oleg	Piater, Justus
Malisiewicz, Tomasz	Neumann, Jan	Pietikainen, Matti
Mallick, Satya	Nevatia, Ram	Pinz, Axel
Manduchi, Roberto	Nicolls, Fred	Pless, Robert
Manmatha, R.	Niebles, Juan Carlos	Pock, Thomas
Marchand, Eric	Nielsen, Mads	Poh, Norman
Marcialis, Gian	Nishino, Ko	Pollefeys, Marc
Marks, Tim	Nixon, Mark	Ponce, Jean
Marszalek, Marcin	Nowozin, Sebastian	Pons, Jean-Philippe
Martinec, Daniel	O'donnell, Thomas	Potetz, Brian
Martinez, Aleix	Obozinski, Guillaume	Prabhakar, Salil
Matei, Bogdan	Odobez, Jean-Marc	Qian, Gang
Mateus, Diana	Odone, Francesca	Quattoni, Ariadna
Matsushita, Yasuyuki	Ofek, Eyal	Radeva, Petia
Matthews, Iain	Ogale, Abhijit	Radke, Richard
Maxwell, Bruce	Okabe, Takahiro	Rakotomamonjy, Alain
Maybank, Stephen	Okatani, Takayuki	Ramanan, Deva
Mayer, Helmut	Okuma, Kenji	Ramanathan, Narayanan
McCloskey, Scott	Olson, Clark	Ranzato, Marc'Aurelio
McKenna, Stephen	Olsson, Carl	Raviv, Dan
Medioni, Gerard	Ommer, Bjorn	Reid, Ian
Meer, Peter	Osadchy, Margarita	Reitmayr, Gerhard
Mei, Christopher	Overgaard, Niels	Ren, Xiaofeng
Michael, Nicholas	Christian	Rittscher, Jens
Micusik, Branislav	Ozuysal, Mustafa	Rogez, Gregory
Minh, Nguyen	Pajdla, Tomas	Rosales, Romer
Mirmehdi, Majid	Panagopoulos,	Rosenberg, Charles
Mittal, Anurag	Alexandros	Rosenhahn, Bodo
Miyazaki, Daisuke	Pandharkar, Rohit	Rosman, Guy
Monasse, Pascal	Pankanti, Sharath	Ross, Arun
Mordohai, Philippos	Pantic, Maja	Roth, Peter
Moreno-Noguer,	Papadopoulo, Theo	Rother, Carsten
Francesc	Parameswaran, Vasu	Rothganger, Fred
Mori, Greg	Parikh, Devi	Rougon, Nicolas
Morimoto, Carlos	Paris, Sylvain	Roy, Sebastien
Morse, Bryan	Patow, Gustavo	Rueckert, Daniel
Moses, Yael	Patras, Ioannis	Ruether, Matthias
Mueller, Henning	Pavlovic, Vladimir	Russell, Bryan

- Russell, Christopher  
 Sahbi, Hichem  
 Stiefelhagen, Rainer  
 Saad, Ali  
 Saffari, Amir  
 Salgian, Garbis  
 Salzmann, Mathieu  
 Sangineto, Enver  
 Sankaranarayanan,  
     Aswin  
 Sapiro, Guillermo  
 Sara, Radim  
 Sato, Imari  
 Savarese, Silvio  
 Savchynskyy, Bogdan  
 Sawhney, Harpreet  
 Scharr, Hanno  
 Scharstein, Daniel  
 Schellewald, Christian  
 Schiele, Bernt  
 Schindler, Grant  
 Schindler, Konrad  
 Schlesinger, Dmitrij  
 Schoenemann, Thomas  
 Schroff, Florian  
 Schubert, Falk  
 Schultz, Thomas  
 Se, Stephen  
 Seidel, Hans-Peter  
 Serre, Thomas  
 Shah, Mubarak  
 Shakhnarovich, Gregory  
 Shan, Ying  
 Shashua, Amnon  
 Shechtman, Eli  
 Sheikh, Yaser  
 Shekhovtsov, Alexander  
 Shet, Vinay  
 Shi, Jianbo  
 Shimshoni, Ilan  
 Shokoufandeh, Ali  
 Sigal, Leonid  
 Simon, Loic  
 Singara,ju, Dheeraaj  
 Singh, Maneesh  
 Singh, Vikas  
 Sinha, Sudipta  
 Sivic, Josef  
 Slabaugh, Greg  
 Smeulders, Arnold  
 Sminchisescu, Cristian  
 Smith, Kevin  
 Smith, William  
 Snavelly, Noah  
 Snoek, Cees  
 Soatto, Stefano  
 Sochen, Nir  
 Sochman, Jan  
 Sofka, Michal  
 Sorokin, Alexander  
 Southall, Ben  
 Souvenir, Richard  
 Srivastava, Anuj  
 Stauffer, Chris  
 Stein, Gideon  
 Strecha, Christoph  
 Sugimoto, Akihiro  
 Sullivan, Josephine  
 Sun, Deqing  
 Sun, Jian  
 Sun, Min  
 Sunkavalli, Kalyan  
 Suter, David  
 Svoboda, Tomas  
 Syeda-Mahmood,  
     Tanveer  
 Süssstrunk, Sabine  
 Tai, Yu-Wing  
 Takamatsu, Jun  
 Talbot, Hugues  
 Tan, Ping  
 Tan, Robby  
 Tanaka, Masayuki  
 Tao, Dacheng  
 Tappen, Marshall  
 Taylor, Camillo  
 Theobalt, Christian  
 Thonnat, Monique  
 Tieu, Kinh  
 Tistarelli, Massimo  
 Todorovic, Sinisa  
 Toreyin, Behcet Ugur  
 Torresani, Lorenzo  
 Torsello, Andrea  
 Toshev, Alexander  
 Trucco, Emanuele  
 Tschumperle, David  
 Tsin, Yanghai  
 Tu, Peter  
 Tung, Tony  
 Turek, Matt  
 Turk, Matthew  
 Tuzel, Oncel  
 Tyagi, Ambrish  
 Urschler, Martin  
 Urtasun, Raquel  
 Van de Weijer, Joost  
 van Gemert, Jan  
 van den Hengel, Anton  
 Vasilescu, M. Alex O.  
 Vedaldi, Andrea  
 Veeraraghavan, Ashok  
 Veksler, Olga  
 Verbeek, Jakob  
 Vese, Luminita  
 Vitaladevuni, Shiv  
 Vogiatzis, George  
 Vogler, Christian  
 Wachinger, Christian  
 Wada, Toshikazu  
 Wagner, Daniel  
 Wang, Chaohui  
 Wang, Hanzi  
 Wang, Hongcheng  
 Wang, Jue  
 Wang, Kai  
 Wang, Song  
 Wang, Xiaogang  
 Wang, Yang  
 Weese, Juergen  
 Wei, Yichen  
 Wein, Wolfgang  
 Welinder, Peter  
 Werner, Tomas  
 Westin, Carl-Fredrik

Wilburn, Bennett	Yang, Peng	Zhang, Cha
Wildes, Richard	Yang, Qingxiong	Zhang, Li
Williams, Oliver	Yang, Ruigang	Zhang, Sheng
Wills, Josh	Ye, Jieping	Zhang, Weiwei
Wilson, Kevin	Yeung, Dit-Yan	Zhang, Wenchao
Wojek, Christian	Yezzi, Anthony	Zhao, Wenyi
Wolf, Lior	Yilmaz, Alper	Zheng, Yuanjie
Wright, John	Yin, Lijun	Zhou, Jinghao
Wu, Tai-Pang	Yoon, Kuk Jin	Zhou, Kevin
Wu, Ying	Yu, Jingyi	Zhu, Leo
Xiao, Jiangjian	Yu, Kai	Zhu, Song-Chun
Xiao, Jianxiong	Yu, Qian	Zhu, Ying
Xiao, Jing	Yu, Stella	Zickler, Todd
Yagi, Yasushi	Yuille, Alan	Zikic, Darko
Yan, Shuicheng	Zach, Christopher	Zisserman, Andrew
Yang, Fei	Zaid, Harchaoui	Zitnick, Larry
Yang, Jie	Zelnik-Manor, Lihi	Zivny, Stanislav
Yang, Ming-Hsuan	Zeng, Gang	Zuffi, Silvia

## Sponsoring Institutions

### Platinum Sponsor

INSTITUT NATIONAL  
DE RECHERCHE  
EN INFORMATIQUE  
ET EN AUTOMATIQUE



### Gold Sponsors



### Silver Sponsors





# Table of Contents – Part V

## Spotlights and Posters W2

Towards Computational Models of the Visual Aesthetic Appeal of Consumer Videos .....	1
<i>Anush K. Moorthy, Pere Obrador, and Nuria Oliver</i>	
Object Recognition Using Junctions .....	15
<i>Bo Wang, Xiang Bai, Xinggang Wang, Wenyu Liu, and Zhuowen Tu</i>	
Using Partial Edge Contour Matches for Efficient Object Category Localization .....	29
<i>Hayko Riemenschneider, Michael Donoser, and Horst Bischof</i>	
Active Mask Hierarchies for Object Detection .....	43
<i>Yuanhao Chen, Long (Leo) Zhu, and Alan Yuille</i>	
From a Set of Shapes to Object Discovery .....	57
<i>Nadia Payet and Sinisa Todorovic</i>	
What Does Classifying More Than 10,000 Image Categories Tell Us? ...	71
<i>Jia Deng, Alexander C. Berg, Kai Li, and Li Fei-Fei</i>	
Modeling and Analysis of Dynamic Behaviors of Web Image Collections .....	85
<i>Gunhee Kim, Eric P. Xing, and Antonio Torralba</i>	
Non-local Characterization of Scenery Images: Statistics, 3D Reasoning, and a Generative Model .....	99
<i>Tamar Avraham and Michael Lindenbaum</i>	
Efficient Highly Over-Complete Sparse Coding Using a Mixture Model .....	113
<i>Jianchao Yang, Kai Yu, and Thomas S. Huang</i>	
Attribute-Based Transfer Learning for Object Categorization with Zero/One Training Example .....	127
<i>Xiaodong Yu and Yiannis Aloimonos</i>	
Image Classification Using Super-Vector Coding of Local Image Descriptors .....	141
<i>Xi Zhou, Kai Yu, Tong Zhang, and Thomas S. Huang</i>	
A Discriminative Latent Model of Object Classes and Attributes .....	155
<i>Yang Wang and Greg Mori</i>	

Seeing People in Social Context: Recognizing People and Social Relationships . . . . . 169  
*Gang Wang, Andrew Gallagher, Jiebo Luo, and David Forsyth*

Discovering Multipart Appearance Models from Captioned Images . . . . . 183  
*Michael Jamieson, Yulia Eskin, Afsaneh Fazly, Suzanne Stevenson, and Sven Dickinson*

Voting by Grouping Dependent Parts . . . . . 197  
*Pradeep Yarlagadda, Antonio Monroy, and Björn Ommer*

Superpixels and Supervoxels in an Energy Optimization Framework . . . . . 211  
*Olga Veksler, Yuri Boykov, and Paria Mehrani*

**Segmentation**

Convex Relaxation for Multilabel Problems with Product Label Spaces . . . . . 225  
*Bastian Goldluecke and Daniel Cremers*

Graph Cut Based Inference with Co-occurrence Statistics . . . . . 239  
*Lubor Ladicky, Chris Russell, Pushmeet Kohli, and Philip H.S. Torr*

Ambrosio-Tortorelli Segmentation of Stochastic Images . . . . . 254  
*Torben Pätz and Tobias Preusser*

Multiple Hypothesis Video Segmentation from Superpixel Flows . . . . . 268  
*Amelio Vazquez-Reina, Shai Avidan, Hanspeter Pfister, and Eric Miller*

Object Segmentation by Long Term Analysis of Point Trajectories . . . . . 282  
*Thomas Brox and Jitendra Malik*

**Spotlights and Posters R1**

Exploiting Repetitive Object Patterns for Model Compression and Completion . . . . . 296  
*Luciano Spinello, Rudolph Triebel, Dizan Vasquez, Kai O. Arras, and Roland Siegwart*

Feature Tracking for Wide-Baseline Image Retrieval . . . . . 310  
*Ameesh Makadia*

Crowd Detection with a Multiview Sampler . . . . . 324  
*Weina Ge and Robert T. Collins*

A Unified Contour-Pixel Model for Figure-Ground Segmentation . . . . . 338  
*Ben Packer, Stephen Gould, and Daphne Koller*

SuperParsing: Scalable Nonparametric Image Parsing with Superpixels . . . . .	352
<i>Joseph Tighe and Svetlana Lazebnik</i>	
Segmenting Salient Objects from Images and Videos . . . . .	366
<i>Esa Rahtu, Juho Kannala, Mikko Salo, and Janne Heikkilä</i>	
ClassCut for Unsupervised Class Segmentation . . . . .	380
<i>Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari</i>	
A Dynamic Programming Approach to Reconstructing Building Interiors . . . . .	394
<i>Alex Flint, Christopher Mei, David Murray, and Ian Reid</i>	
Discriminative Mixture-of-Templates for Viewpoint Classification . . . . .	408
<i>Chunhui Gu and Xiaofeng Ren</i>	
Efficient Non-consecutive Feature Tracking for Structure-from-Motion . . . . .	422
<i>Guofeng Zhang, Zilong Dong, Jiaya Jia, Tien-Tsin Wong, and Hujun Bao</i>	
<i>P2II</i> : A Minimal Solution for Registration of 3D Points to 3D Planes . . . . .	436
<i>Srikumar Ramalingam, Yuichi Taguchi, Tim K. Marks, and Oncel Tuzel</i>	
Boosting Chamfer Matching by Learning Chamfer Distance Normalization . . . . .	450
<i>Tianyang Ma, Xingwei Yang, and Longin Jan Latecki</i>	
Geometry Construction from Caustic Images . . . . .	464
<i>Manuel Finckh, Holger Dammertz, and Hendrik P.A. Lensch</i>	
Archive Film Restoration Based on Spatiotemporal Random Walks . . . . .	478
<i>Xiaosong Wang and Majid Mirmehdi</i>	
Reweighted Random Walks for Graph Matching . . . . .	492
<i>Minsu Cho, Jungmin Lee, and Kyoung Mu Lee</i>	
Rotation Invariant Non-rigid Shape Matching in Cluttered Scenes . . . . .	506
<i>Wei Lian and Lei Zhang</i>	
Loosely Distinctive Features for Robust Surface Alignment . . . . .	519
<i>Andrea Albarelli, Emanuele Rodolà, and Andrea Torsello</i>	
Accelerated Hypothesis Generation for Multi-structure Robust Fitting . . . . .	533
<i>Tat-Jun Chin, Jin Yu, and David Suter</i>	

Aligning Spatio-Temporal Signals on a Special Manifold . . . . .	547
<i>Ruonan Li and Rama Chellappa</i>	
Supervised Label Transfer for Semantic Segmentation of Street Scenes . . . . .	561
<i>Honghui Zhang, Jianxiong Xiao, and Long Quan</i>	
Category Independent Object Proposals . . . . .	575
<i>Ian Endres and Derek Hoiem</i>	
Photo-Consistent Planar Patches from Unstructured Cloud of Points . . .	589
<i>Roberto Toldo and Andrea Fusiello</i>	
Contour Grouping and Abstraction Using Simple Part Models . . . . .	603
<i>Pablo Sala and Sven Dickinson</i>	
Dynamic Color Flow: A Motion-Adaptive Color Model for Object Segmentation in Video . . . . .	617
<i>Xue Bai, Jue Wang, and Guillermo Sapiro</i>	
What Is the Chance of Happening: A New Way to Predict Where People Look . . . . .	631
<i>Yezhou Yang, Mingli Song, Na Li, Jiajun Bu, and Chun Chen</i>	
Supervised and Unsupervised Clustering with Probabilistic Shift . . . . .	644
<i>Sanketh Shetty and Narendra Ahuja</i>	
Depth-Encoded Hough Voting for Joint Object Detection and Shape Recovery . . . . .	658
<i>Min Sun, Gary Bradski, Bing-Xin Xu, and Silvio Savarese</i>	
Shape Analysis of Planar Objects with Arbitrary Topologies Using Conformal Geometry . . . . .	672
<i>Lok Ming Lui, Wei Zeng, Shing-Tung Yau, and Xianfeng Gu</i>	
A Coarse-to-Fine Taxonomy of Constellations for Fast Multi-class Object Detection . . . . .	687
<i>Sanja Fidler, Marko Boben, and Aleš Leonardis</i>	
Object Classification Using Heterogeneous Co-occurrence Features . . . . .	701
<i>Satoshi Ito and Susumu Kubota</i>	
Converting Level Set Gradients to Shape Gradients . . . . .	715
<i>Siqi Chen, Guillaume Charpiat, and Richard J. Radke</i>	
A Close-Form Iterative Algorithm for Depth Inferring from a Single Image . . . . .	729
<i>Yang Cao, Yan Xia, and Zengfu Wang</i>	

Learning Shape Segmentation Using Constrained Spectral Clustering and Probabilistic Label Transfer .....	743
<i>Avinash Sharma, Etienne von Lavante, and Radu Horaud</i>	
Weakly Supervised Shape Based Object Detection with Particle Filter .....	757
<i>Xingwei Yang and Longin Jan Latecki</i>	
Geodesic Shape Retrieval via Optimal Mass Transport .....	771
<i>Julien Rabin, Gabriel Peyré, and Laurent D. Cohen</i>	
<b>Spotlights and Posters R2</b>	
Image Segmentation with Topic Random Field .....	785
<i>Bin Zhao, Li Fei-Fei, and Eric P. Xing</i>	
<b>Author Index</b> .....	799

# Towards Computational Models of the Visual Aesthetic Appeal of Consumer Videos

Anush K. Moorthy\*, Pere Obrador, and Nuria Oliver

Telefonica Research, Barcelona, Spain

**Abstract.** In this paper, we tackle the problem of characterizing the aesthetic appeal of consumer videos and automatically classifying them into high or low aesthetic appeal. First, we conduct a controlled user study to collect ratings on the aesthetic value of 160 consumer videos. Next, we propose and evaluate a set of low level features that are combined in a hierarchical way in order to model the aesthetic appeal of consumer videos. After selecting the 7 most discriminative features, we successfully classify aesthetically appealing *vs.* aesthetically unappealing videos with a 73% classification accuracy using a support vector machine.

**Keywords:** Video aesthetics, video quality, subjective assessment.

## 1 Introduction

In today's digital world, we face the challenge of developing efficient multimedia data management tools that enable users to organize and search multimedia content from growing repositories of digital media. Increasing storage capabilities at low prices combined with pervasive devices to capture digital images and videos enable the generation and archival of unprecedented amounts of personal multimedia content. For example, as of May 2009, about 20 hours of video footage – most of it user-generated – were uploaded on the popular video sharing site YouTube every minute [1]. In addition, the number of user-generated video creators is expected to grow in the US by 77% from 2008 to 2013 [2].

Text query-based image and video search approaches rely heavily on the similarity between the input textual query and the textual metadata (*e.g.* tags, comments, etc.) that has previously been added to the content by users. Relevance is certainly critical to the satisfaction of users with their search results, yet not sufficient. For example, any visitor of YouTube will attest to the fact that the most *relevant* search results today include a large amount of user generated data of *varying aesthetic quality*, where aesthetics deal with the human appreciation of beauty. Hence, filtering and re-ranking the videos with a measure of their aesthetic value would probably improve the user experience and satisfaction with the search results. In addition to improving search results, another

---

\* A. K. Moorthy is with The University of Texas at Austin, Austin, Texas, USA - 78712. This work was performed when A. K. Moorthy was an intern at Telefonica Research, Barcelona, Spain.

challenge faced by video sharing sites is being able to attract advertisement to the user generated content, particularly given that some of it is deemed to be “unwatchable” [3], and advertisers are typically reluctant to place their clients’ brands next to any material that may damage their clients’ reputations [4]. We believe that the analysis of the aesthetic value of videos may be one of the tools used to automatically identify the material that is “advertisement worthy” *vs.* not. Last, but not least, video management tools that include models of aesthetic appeal may prove very useful to help users navigate and enjoy their ever increasing – yet rarely seen – personal video collections.

Here, we focus on *building computational models of the aesthetic appeal of consumer videos*. Note that video aesthetic assessment differs from video quality assessment (VQA) [5] in that the former seeks to evaluate the holistic appeal of a video and hence encompasses the latter. For example, a low quality video with severe blockiness will have low aesthetic appeal. However, a poorly lit undistorted video with washed-out colors may have high quality but may also be aesthetically unappealing. Even though image aesthetic assessment has recently received the attention of the research community [6,7,8,9,10], video aesthetic assessment remains little explored [8].

To the best of our knowledge, the work presented in this paper represents the first effort to automatically characterize the aesthetic appeal of *consumer* videos and classify them into high or low aesthetic appeal. For this purpose, we first carry out a controlled user study (Section 3) to collect unbiased estimates of the aesthetic appeal of 160 consumer videos and thus generate ground truth. Next, we propose low-level features calculated on a per-frame basis, that are correlated to visual aesthetics (Section 4.1), followed by novel strategies to combine these frame-level features to yield video-level features (Section 4.2). Note that previous work in this area has simply used the mean value of each feature across the video [8], which fails to capture the video dynamics and the peculiarities associated with human perception [11]. Finally, we evaluate the proposed approach with the collected 160 videos, compare our results with the state-of-the-art (Section 5), discuss the implications of our findings (Section 6) and highlight our lines of future work (Section 7).

In sum, the main contributions of this paper are threefold: 1) We carry out a controlled user study to collect unbiased ground-truth about the aesthetic appeal of 160 consumer videos; 2) we propose novel low-level (*i.e.*, frame-level) and video-level features to characterize video aesthetic appeal; and 3) we quantitatively evaluate our approach, compare our results with the state-of-the-art and show how our method is able to correctly classify videos into low or high aesthetic appeal with 73% accuracy.

## 2 Previous Work

**Aesthetic Appeal in Still Images:** One of the earliest works in this domain is that by Savakis *et al.* [12] where they performed a large scale study of the possible features that might have an influence on the aesthetic rating of an image. However, no algorithm was proposed to evaluate appeal. In [10], Tong *et al.*

extracted features – including measures of color, energy, texture and shape – from images and a two-class classifier (high *vs.* low aesthetic appeal) was proposed and evaluated using a large image database with photos from COREL and Microsoft Office Online (high aesthetic appeal) and from staff at Microsoft Research Asia (low aesthetic appeal). One drawback with this approach is that some of the selected features lacked photographic/perceptual justification. Furthermore, their dataset assumed that home users are poorer photographers than professionals, which may not always be true.

Datta *et al.* [6] extracted a large set of features based on photographic rules. Using a dataset from an online image sharing community, the authors discovered the top 15 features in terms of their cross validation performance with respect to the image ratings. The authors reported a classification (high *vs.* low aesthetic appeal) accuracy of 70.12%. Ke *et al.* [7] utilized a top-down approach, where a small set of features based on photographic rules were extracted. A dataset obtained by crawling DPChallenge.com was used and the photo’s average rating was utilized as ground truth. In [8], Luo and Tang furthered the approach proposed in [7] by extracting the main subject region (using a sharpness map) in the photograph. A small set of features were tested on the same database as in [7], and their approach was shown to perform better than that of Datta *et al.* [6] and Ke *et al.* [7]. Finally, Obrador recently proposed a region-of-interest based approach to compute image aesthetic appeal [9] where the region-of-interest is extracted using a combination of sharpness, contrast and colorfulness. The size of the region-of-interest, its isolation from the background and its exposure were then computed to quantify aesthetic appeal with good results on a photo dataset created by the author.

**Aesthetic Appeal in Videos:** To the best of our knowledge, only the work in [8] has tackled the challenge of modeling video aesthetics, in which their goal was to automatically distinguish between low quality (amateurish) and high quality (professional) videos. They applied image aesthetic measures – where each feature was calculated on a subset of the video frames at a rate of 1 frame per second (fps) – coupled with two video-specific features (length of the motion of the main subject region and motion stability). The mean value of each feature across the whole video was utilized as the video representation. They evaluated their approach on a large database of YouTube videos and achieved good classification performance of professional *vs.* amateur videos ( $\approx 95\%$  accuracy).

### 3 Ground Truth Data Collection

Previous work in the field of image aesthetics has typically used images from online image-sharing websites [13]. Each of these photo-sharing sites allows users to rate the images, but not necessarily according to their aesthetic appeal. A few websites (*e.g.* Photo.net) do have an aesthetic scale (1-7) on which users rate the photographs. However, the lack of a controlled test environment implies that the amount of noise associated with the ratings in these datasets is typically large [14]. In addition, users are influenced in their aesthetic ratings by factors



such as the artist who took the photograph, the relation of the subject to the photographer, the content of the scene and the context under which the rating is performed. Hence, a controlled study to collect aesthetic rating data is preferred over ratings obtained from a website. As noted in [13], web-based ratings are mainly used due to a lack of controlled experimental ground truth data on the aesthetic appeal of images or videos. In the area of image aesthetics, we shall highlight two controlled user studies [9,12], even though neither of these datasets was made public.

To the best of our knowledge, the only dataset in the area of video aesthetics is that used by Luo and Tang [8]. It consists of 4000 high quality (professional) and 4000 low quality (amateurish) YouTube videos. However, the authors do not explain how the dataset was obtained or how the videos were ranked. The number of subjects that participated in the ranking is unknown. It is unclear if the videos were all of the same length. Note that the length of the video has been shown to influence the ratings [15]. The content of the videos is unknown and since the rating method is undisclosed, it is unclear if the participants were influenced by the content when providing their ratings. Finally, the authors do not specify if the rated videos had audible audio or not. It is known that the presence of audio influences the overall rating of a video [16].

In order to address the above mentioned drawbacks and to create a publicly available dataset for further research, we conducted a controlled user study where 33 participants rated the aesthetic appeal of 160 videos<sup>1</sup>. The result of the study is a collection of 160 videos with their corresponding aesthetic ratings which was used as ground truth in our experiments. In this section, we detail how the videos were selected and acquired, and how the study was conducted.

**Video Selection:** Since the focus of our work is consumer videos, we crawled the YouTube categories that were more likely to contain consumer generated content: Pets & Animals, Travel & Events, Howto & Style, and so on. To collect the videos, we used popular YouTube queries from the aforementioned categories (*i.e.*, text associated with the most viewed videos in those categories), for instance, “puppy playing with ball” and “baby laughing”. In addition and in order to have a wide diversity of video types, we included semantically different queries that retrieved large numbers (>1000) of consumer videos, such as “Rio de Janeiro carnival” and “meet Mickey Mouse Disney”. In total, we downloaded 1600 videos (100 videos  $\times$  16 queries). A 15 second segment was extracted from the middle part of each of the videos in order to reduce potential biases induced by varying video lengths [15]. Each of the 1600 videos was viewed by two of the authors who rated the aesthetic appeal of the videos on a 5-point Likert scale. The videos that were not semantically relevant to the search query were discarded (*e.g.*, “puppy playing with ball” produced videos which had children and puppies playing together or just children playing together); videos that were professionally generated were also discarded. A total of 992 videos were retained from the initial 1600. Based on the mean ratings of the videos – from the two

---

<sup>1</sup> Each video received 16 different ratings by a subset of 16 participants.

sets of scores by the authors after converting them to Z-scores [17], 10 videos were picked for each query such that they uniformly covered the 5-point range of aesthetic ratings. Thus, a total of 160 videos – 10 videos  $\times$  16 queries – were selected for the study. The selected videos were uploaded to YouTube to ensure that they would be available for the study and future research.

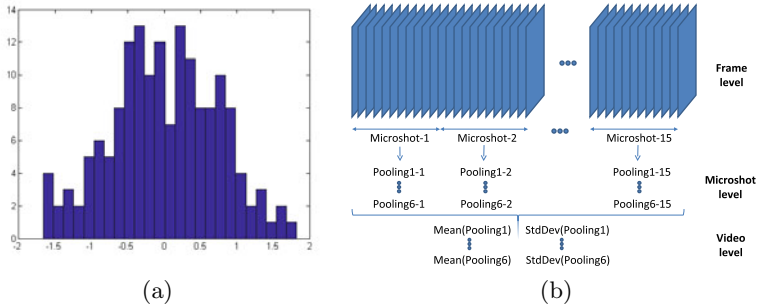
**User Study:** An important reason for conducting a controlled study is the role that content (*i.e.*, "what" is recorded in the video) plays in video ratings. As noted in [13], the assessment of videos is influenced by both their *content* and their *aesthetic* value. We recognize that these two factors are not completely independent of each other. However in order to create a content-independent algorithm that relies on low-level features to measure the aesthetic value of a video, the ground truth study design must somehow segregate these two factors. Hence, our study required users to rate the videos on two scales: *content* and *aesthetics*, in order to reduce the influence of the former in the latter.

A total of 33 participants (25 male) took part in the study. They had been recruited by email advertisement in a large corporation. Their ages ranged from 24 to 45 years ( $\mu = 29.1$ ) and most participants were students, researchers or programmers. All participants were computer savvy and 96.8 % reported regularly using video sharing sites such as YouTube. The participants were not tested for acuity of vision, but a verbal confirmation of visual acuity was obtained. Participants were not paid for their time, but they were entered in a \$USD 150 raffle. The study consisted of 30 minute rating sessions where participants were asked to rate both the *content* and the *aesthetic* appeal of 40 videos (10 videos  $\times$  4 queries). Subjects were allowed to participate in no more than two rating sessions (separated by at least 24 hours).

The first task in the study consisted of a short training session involving 10 videos from a "dance" query; the data collected during this training session was not used for the study. The actual study followed. The order of presentation of queries for each subject followed a Latin-square pattern in order to avoid presentation biases. In addition, the order in which the videos were viewed within each query was randomized. The videos were displayed in the center of a 17-inch LCD screen with a refresh rate of 60 Hz and a resolution of 1024  $\times$  768 pixels, on a mid-gray background, and at a viewing distance of 5 times the height of the videos [18]. Furthermore, since our focus is *visual* appeal, the videos were shown without any audio [16].

Before the session began, each participant was instructed as follows. *You will be shown a set of videos on your screen. Each video is 15 seconds long. You have to rate the video on two scales: Content and Aesthetics from very bad (-2) to very good (+2). By content we mean whether you liked the activities in the video, whether you found them cute or ugly for example.*<sup>2</sup> *You are required to watch each video entirely before rating it.* We were careful not to bias participants toward any particular low-level measure of aesthetics. In fact, we left the definition fairly

<sup>2</sup> Each video was embedded into the web interface with two rating scales underneath: one for *content* and the other for *aesthetics*. The scales were: Very Bad (-2), Bad (-1), Fair (0), Good (1), Very Good (2).



**Fig. 1.** (a) Histogram of aesthetic MOS from the user study. (b) Proposed 2-level pooling approach, from frame to microshot (level 1) and video (level 2) features.

open in order to allow participants to form their own opinion on what parameters they believed video aesthetics should be rated on.

During the training session, participants were allowed to ask as many questions as needed. Most questions centered around our definition of *content*. In general, subjects did not seem to have a hard time rating the aesthetics of the videos. At the end of each query, participants were asked to describe in their own words the reasons for their aesthetic ratings of the videos. With this questionnaire, we aimed to capture information about the low-level features that they were using to rate video aesthetics in order to guide the design of our low-level features. Due to space constraints, we leave the analysis of the participants’ answers to these questions for future work.

The study yielded a total of 16 different ratings (across subjects) of video aesthetics for each of the 160 videos. A single per-video visual aesthetic appeal score was created: First, the scores of each participant were normalized by subtracting the mean score per participant and per session from each of the participant’s scores, in order to reduce the bias of the ratings in each session. Next, the average score per video and across all participants was computed to generate a mean opinion score (MOS). This approach is similar to that followed for Z-scores [17]. Thus, a total of 160 videos with ground truth about their aesthetic appeal in the form of MOS were obtained. Figure 1 (a) depicts the histogram of the aesthetic MOS for the 160 videos, where 82 videos were rated below zero, and 78 videos were rated above zero. Even though 160 videos may seem small compared to previous work [8], datasets of the same size are common in state-of-the-art controlled user studies of video quality assessment [19].

## 4 Feature Computation

The features presented here were formulated based on previous work, the feedback from our user study and our own intuition.

The main difference between an image and a video is the presence of the temporal dimension. In fact, humans do not perceive a series of images in the

same fashion as they perceive a video [5]. Hence, the features to be extracted from the videos should incorporate information about this temporal dimension. In this paper, we propose a hierarchical *pooling* approach to collapse each of the features extracted on a frame-by-frame basis into a single value for the entire video, where *pooling* [11] is defined as the process of collapsing a set of features, either spatially or temporally. In particular, we perform a two-level *pooling* approach, as seen in Fig. 1 (b). First, basic features are extracted on a frame-by-frame basis. Next, the frame-level features are pooled within each microshot<sup>3</sup> using 6 different pooling techniques, generating 6 microshot-level features for each basic feature. Finally, the microshot-level features are pooled across the entire video using two methods (mean and standard deviation), thus generating a set of 12 video-level features for each of the basic frame-level features.

In the following sections we describe the basic frame-level features and their relationship (if any) to previous work, followed by the hierarchical pooling strategy used to collapse frame-level values into video-level descriptors.

#### 4.1 Frame-Level Features

**Actual Frame Rate** ( $f_1$ , actual-fps): 29% of the downloaded videos contained repeated frames. In an extreme case, a video which claimed to have a frame-rate of 30 fps had an actual new frame every 10 repetitions of the previous frame. Since frame-rate is an integral part of perceived quality [5] – and hence aesthetics, our first feature,  $f_1$ , is the “true” frame-rate of the video. In order to detect frame repetition, we use the structural similarity index (SSIM) [20].

A measure of the perceptual similarity of consecutive frames is given by  $Q = 1 - SSIM$  (small  $Q$  indicates high similarity), and is computed between neighboring frames creating a vector  $\mathbf{m}$ . To measure periodicity due to frame insertions, we compute  $\mathbf{m}^{th} = \{ind(m_i) | m_i \leq 0.02\}$ , where the set threshold allows for a small amount of dissimilarity between adjacent frames (due to encoding artifacts). This signal is differentiated (with a first order filter  $h[i] = [1 - 1]$ ) to obtain  $d\mathbf{m}$ . If this is a periodic signal then we conclude that frames have been inserted, and the true frame rate is calculated as:  $f_1 = fps \times \frac{MAX(d\mathbf{m})-1}{T_m}$ , where  $T_m$  is the number of samples in  $\mathbf{m}$  corresponding to the period in  $d\mathbf{m}$ . Note that this feature has not been used before to assess video aesthetics.

**Motion Features** ( $f_2$ , motion-ratio, and  $f_3$ , size-ratio): The human visual system devotes a significant amount of resources for motion processing. Jerky camera motion, camera shake and fast object motion in video are distracting and they may significantly affect the aesthetic appeal of the video. While other authors have proposed techniques to measure shakiness in video [21], our approach stems from the hypothesis that a good consumer video contains two regions: the foreground and the background. We further hypothesize that the ratio of motion magnitudes between these two regions and their relative sizes have a direct impact on video aesthetic appeal.

<sup>3</sup> In our implementation a microshot is a set of frames amounting to one second of video footage.

A block-based motion estimation algorithm is applied to compute motion vectors between adjacent frames. Since the videos in our set are compressed videos from YouTube, blocking artifacts may hamper the motion estimates. Hence, motion estimation is performed after low-pass filtering and downsampling by 2 in each dimension, each video frame. For each pixel location in a frame, the magnitude of the motion vector is computed. Then, a k-means algorithm with 2 clusters is run in order to segregate the motion vectors into two classes. Within each class, the motion vector magnitudes are histogrammed and the magnitude of the motion vector corresponding to the peak of the histogram is chosen as a representative vector for that class. Let  $m_f$  and  $m_b$  denote the magnitude of the motion vectors for each of the classes, where  $m_f > m_b$ , and let  $s_f$  and  $s_b$  denote the size (in pixels) of each of the regions respectively. We compute  $f_2 = \frac{m_b+1}{m_f+1}$  and  $f_3 = \frac{s_b+1}{s_f+1}$ . The constant 1 is added in order to prevent numerical instabilities in cases where the magnitude of motion or size tends to zero. These features have not been used before to characterize video aesthetics.

**Sharpness/Focus of the Region of Interest** ( $f_4$ , focus): Sharpness is of utmost importance when assessing visual aesthetics [9]. Note that our focus lies in consumer videos where the cameras are typically focused at optical infinity, such that measuring regions in focus is challenging. In order to extract the in-focus region, we use the algorithm proposed in [22] and set the median of the level of focus of the ROI as our feature  $f_4$ .

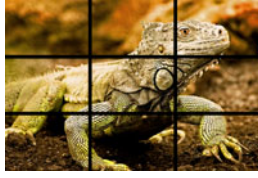
**Colorfulness** ( $f_5$ , colorfulness): Videos which are colorful tend to be seen as more attractive than those in which the colors are “washed out” [23]. The colorfulness of a frame ( $f_5$ ) is evaluated using the technique proposed in [23]. This measure has previously been used in [9] to quantify the aesthetics of images.

**Luminance** ( $f_6$ , luminance): Luminance has been shown to play a role in the aesthetic appeal of images [6]. Images (and videos) in either end of the luminance scale (*i.e.*, poorly lit or with extremely high luminance) are typically rated as having low aesthetic value [4]. Hence, we compute the luminance feature  $f_6$  as the mean value of the luminance within a frame.

**Color Harmony** ( $f_7$ , harmony): The colorfulness measure does not take into account the effect that the combination of different colors has on the aesthetic value of each frame. To this effect, we evaluate color harmony using a variation of the technique by Cohen-Or *et al.* [24] where they propose eight harmonic types or templates over the hue channel in the HSV space. Note that one of these templates (N-type) corresponds to grayscale images and hence does not apply to the videos in our study. We compute the (normalized) hue-histogram of each frame and convolve this histogram with each of the 7 templates [5]. The peak of the convolution is selected as a measure of similarity of the frame’s histogram to a particular template. The maximum value of these 7 harmony similarity measures

<sup>4</sup> A video with alternating low and high luminance values may also have low aesthetic appeal.

<sup>5</sup> The template definitions are the same as the ones proposed in [24].



**Fig. 2.** Rule of thirds: the head of the iguana is placed in the top-right intersecting point

(one for each template) is chosen as our color harmony feature. Other color harmony measures have been used to assess the aesthetic quality of paintings [25], and photos and video [8].

**Blockiness Quality** ( $f_8$ , quality): The block-based approach used in current video compression algorithms leads to the presence of blocking artifacts in videos. Blockiness is an important aspect of quality and for compressed videos it has been shown to overshadow other artifacts [26]. The YouTube consumer videos from our dataset are subject to video compression and hence we evaluate their quality by looking for blocking artifacts as in [26]. Since this algorithm was proposed for JPEG compression, it is defined for  $8 \times 8$  blocks only. However, some YouTube videos are compressed using H.264/AVC which allows for multiple block sizes [27]. Hence, we modified the algorithm in [26] to account for multiple block sizes. In our experiments, however, we found that different block sizes did not improve the performance of the quality feature. Therefore, in our evaluation we use the  $8 \times 8$  block-based quality assessment as in [26] and denote this quality feature as  $f_8$ . We are not aware of any previously proposed aesthetic assessment algorithm that includes a blockiness quality measure.

**Rule of thirds** ( $f_9$ , thirds): One feature that is commonly found in the literature on aesthetics and in books on professional photography is the rule of thirds [28]. This rule states that important compositional elements of the photograph should be situated in one of the four possible *power points* in an image (*i.e.*, in one of the four intersections of the lines that divide the image into nine equal rectangles, as seen in Figure 2). In order to evaluate a feature corresponding to the rule of thirds, we utilize the region of interest (ROI) extracted as described above. Similarly to [8], our measure of the rule of thirds ( $f_9$ ) is the minimum distance of the centroid of the ROI to these four points.

## 4.2 Microshot and Video-Level Features

Once the 8 frame-level features ( $f_2$  to  $f_9$ ) have been computed on every frame, they are combined to generate features at the microshot (*i.e.*, 1 second of video footage) level which are further combined to yield features at the video level.

We compute 6 different feature pooling techniques for each basic frame level feature – *mean*, *median*, *min*, *max*, *first quartile* (labeled as *fourth*) and *third quartile* (labeled as *three-fourths*) – in order to generate the microshot-level

features, and we let our classifier automatically select the most discriminative features. In this paper we pool microshot-level features with two strategies in order to generate video-level features: *average*, computed as the mean (labeled as *mean*) of the features across all microshots; and standard deviation (labeled as *std*), again computed across all microshots in the video. Thus, a bag of 97 video-level features is generated for each video: 8 frame-level basic features  $\times$  6 pooling techniques at the microshot level  $\times$  2 pooling techniques at the video level  $+ f_1$ .

In the remainder of the paper, we shall use the following nomenclature: *videoLevel-microshotLevel-basicFeature*, to refer to each of the 97 features. For example, the basic feature *harmony* ( $f_7$ ), pooled using the median at the microshot level and the mean at the video level would be referred as: *mean-median-harmony*. The use of these pooling techniques is one of the main contributions of this paper. Previous work [8] has only considered a downsampling approach at the microshot level (at 1 fps), and an averaging pooling technique at the video level, generating one single video level feature for each basic feature, which cannot model their temporal variability.

## 5 Experimental Results

Even though one may seek to automatically estimate the aesthetic ratings of the videos, the subjectivity of the task makes it a very difficult problem to solve [13]. Therefore, akin to previous work in this area, we focus on automatically classifying the videos into two categories: aesthetically appealing *vs.* aesthetically unappealing. The ground truth obtained in our user study is hence split into these two categories, where the median of the aesthetic scores is considered as the threshold. All scores above the median value are labeled as *appealing* (80 videos) and those below are labeled as *unappealing* (80 videos). In order to classify the videos into these two classes, we use a support vector machine (SVM) [29] with a radial basis function (RBF) kernel ( $C, \gamma = (1, 3.7)$ ) and the LibSVM package [30] for implementation.

We perform a five-fold cross-validation where 200 train/test runs are carried out with the feature sets that are being tested. We first evaluate the classification performance of each of the 97 video-level features individually. The best performing 14 features in these cross-validation tests are shown in Table II. The classification performance of these features is fairly stable: the average standard deviation of the classification accuracy across features and over the 200 runs is 2.1211 (min = 0.5397, max = 3.2779).

In order to combine individual features, we use a hybrid of a filter-based and wrapper-based approach, similar to [6]. We only consider the video-level features that individually perform above 50%. We first pick the video-level feature which classifies the data the best. All the other video-level features derived from the same basic feature and pooled with the same video-level pooling method (*i.e.*, either mean or standard deviation) are discarded from the bag before the next feature is selected. The next selected feature is the one that

classifies the data the best *in conjunction with* the first selected feature, and so on. A 7-dimensional feature vector<sup>6</sup> is thus formed. The selected features in order of their classification performance after being combined with the previously selected features are: actual fps (acc=58.8%,  $\sigma = 1.5$ ); mean-three-fourth-colorfulness (acc=67%,  $\sigma = 1.8$ ); std-median-thirds (acc=69.5%,  $\sigma = 1.9$ ); mean-fourth-focus (acc=69.6%,  $\sigma = 2.2$ ); mean-max-luminance (acc=71%,  $\sigma = 1.9$ ); mean-fourth-quality (acc=72.0%,  $\sigma = 1.9$ ); and std-median-focus (acc=73.0%,  $\sigma = 2.0$ ).

An overall classification accuracy of **73.03%** is thus obtained. In order to provide a comparison with previous work, we implemented the algorithm proposed in [8], achieving a classification accuracy of 53.5%. The poor performance of this algorithm may be attributed to the fact that it was designed for professional *vs.* amateur video classification rather than for classifying consumer videos into high or low visual aesthetic appeal.

**Table 1.** Individual classification accuracy of the top 14-features in descending order of performance

Feature	Accura.	Feature	Accura.
1. actual-fps	58.77%	8. mean-mean-colorfulness	56.34%
2. mean-max-size-ratio	58.68%	9. mean-med-colorfulness	56.21%
3. std-fourth-motion-ratio	58.06%	10. mean-mean-quality	55.73%
4. mean-fourth-quality	57.67%	11. mean-three-fourth-quality	55.70%
5. mean-three-fourth-colorfulness	56.86%	12. mean-max-luminance	55.62%
6. mean-max-colorfulness	56.80%	13. std-three-fourth-motion-ratio	55.19%
7. mean-max-quality	56.62%	14. mean-three-fourth-luminance	55.16%

**Personalization:** Personalization has not been explored before in this area even though it is known that certain aspects of aesthetic sensitivities depend on individual factors [13]. In this section, we carry out a preliminary analysis of the personalization of aesthetic ratings. Recall that two of the authors rated the aesthetic value of 1600 videos. All videos which were semantically irrelevant or professionally generated were excluded from the analysis (608 videos or 38%). Video-level features were computed for the remaining 992 videos. Using the 7-dimensional feature vector previously described, we obtain classification accuracies of 61.66% (author 1) and 58.17% (author 2).

In order to evaluate the impact that personalization would have on this dataset, we select the optimum feature combination – using the approach described above – for each of the authors. Table 2 depicts the selected features and their contributions to classification accuracy, yielding classification accuracies of 63.24% (author 1) and 66.46% (author 2), significantly larger in the case of author 2 than the accuracies obtained with the *non-personalized* feature vector.

<sup>6</sup> The feature vector is restricted to 7-dimensions due to the relatively small number of videos in the ground truth (160) and in order to prevent overfitting.



**Table 2.** Classification accuracies with personalized feature vectors. Features selected for each author and their contribution to accuracy - '+' indicates that the result was obtained by combining this feature with the one right above it.

Author 1			Author 2		
Feature	Accura.	StdDev	Feature	Accura.	StdDev
actual-fps	58.4%	0.1	mean-fourth-luminance	58.0%	0.2
+ mean-mean-quality	60.2%	0.3	+ mean-max-harmony	62.1%	0.5
+ mean-mean-size-ratio	61.2%	0.4	+ std-max-quality	64.1%	0.6
+ mean-fourth-harmony	62.3%	0.7	+ mean-median-size-ratio	65.0%	0.5
+ std-max-quality	63.2%	0.7	+ mean-fourth-focus	66.0%	0.7
+ std-max-size-ratio	63.1%	0.7	+ std-fourth-size-ratio	66.1%	0.6
+ mean-max-luminance	63.1%	0.8	+ mean-max-thirds	66.4%	0.6
+ std-fourth-thirds	63.2 %	0.9	+ std-mean-focus	66.5%	0.7

**Aesthetics vs. Quality:** As we mentioned in the introduction, *quality* does not capture all aspects of the aesthetic appeal of a video, but a holistic definition of aesthetics must include the quality of a video. In order to illustrate the role that quality plays on aesthetics, we evaluate the performance of the quality features – blockiness quality ( $f_8$ ) and actual frames-per-second ( $f_1$ ) – on the aesthetics classification. Hence, a *quality feature vector* is created by combining the actual fps measure ( $f_1$ ) and the blocking quality pooling strategy that gives the best performance (mean-fourth-quality). This vector when used for classification yields an accuracy of 58.0%, which suggests that even though quality is an integral part of aesthetics, the aesthetic value of a video encompasses elements beyond traditional measures of quality. When adding the focus feature ( $f_4$ ), arguably a quality feature also (particularly the *std-median-focus* feature) the overall performance increases to 60.0%, still well below the performance obtained when using the best performing 3 aesthetics features: 69.5%, as previously explained.

## 6 Discussion

Apart from the actual-fps feature ( $f_1$ ), the rest of the features that were automatically selected to classify the aesthetic value of videos correlate well with previous research and intuition. For example, the third quartile of the colorfulness feature ( $f_5$ ) would indicate that the maximum colorfulness value is probably noise, and the statistical measure of third quartile is a stable indicator of colorfulness. Again, the first quartile of the quality feature ( $f_8$ ) correlates with research in image quality assessment [11]. Furthermore, quality features alone do not seem to capture all the elements that characterize the aesthetic value of consumer videos.

The standard deviation of the focus feature ( $f_4$ ) is again intuitive in the sense that humans tend to be more sensitive to changes in focus rather than its absolute value. This is also true for the rule-of-thirds feature ( $f_9$ ), which is a measure of how well the main subject is framed in the video. Even though the motion

features that we computed were not selected in the final feature vector, on their own these features performed well (see Table 1) and seemed to be useful for personalization (Table 2). Given that the number of videos in the personalization dataset is large and that motion features on their own seem to correlate well with perception, we hypothesize that increasing the number of videos in the current dataset (which we plan to undertake in the future) will result in a selection of the motion features as well.

## 7 Conclusions and Future Work

In this paper, we have proposed a hierarchical approach to characterize the aesthetic appeal of consumer videos and automatically classify them into high or low aesthetic appeal. We have first conducted a controlled user study to collect human ratings on the aesthetic value of 160 consumer videos. Next, we have proposed 9 low-level features to characterize the aesthetic appeal of the videos. In order to generate features at the video level, we have proposed and evaluated various pooling strategies (at the microshot and video levels) based on statistical measures. Based on the collected ground truth ratings, we have automatically selected 7 features at the video-level and have classified the videos into high *vs.* low aesthetic appeal with 73% classification accuracy, compared to 53.5% classification accuracy of a state-of-the-art algorithm. The videos and the subjective ratings have been made available publicly<sup>7</sup>.

We plan on increasing the number of videos in our ground truth database and conduct a larger scale user study. Future work includes exploring temporal models to characterize video aesthetics, investigating personalization techniques and shedding light on which features of our aesthetics model may be universal *vs.* person-dependent, and assessing the influence of audio in aesthetic ratings so as to form a complete measure of audio-visual aesthetics. Finally, we also plan to develop novel aesthetics-assisted hierarchical user interfaces to allow end users to efficiently navigate their personal video collections.

## References

1. Junee, R.: 20 Hours of Video Uploaded Every Minute! (2009), <http://youtube-global.blogspot.com/>
2. Verna, P.: A Spotlight on UGC Participants (2009), <http://www.emarketer.com/Article.aspx?R=1006914>
3. Wayne, B.: (2009), <http://www.businessinsider.com/is-youtube-doomed-2009-4>
4. Messaris, P.: Visual persuasion: the role of images in advertising. Sage Publications Inc., Thousand Oaks (1997)
5. Wang, Z., Sheikh, H.R., Bovik, A.C.: Objective video quality assessment. In: The Handbook of Video Databases: Design and Applications, pp. 1041–1078 (2003)
6. Datta, R., Joshi, D., Li, J., Wang, J.: Studying aesthetics in photographic images using a computational approach. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 288–301. Springer, Heidelberg (2006)

---

<sup>7</sup> <http://mm2.tid.es/videoAestheticsUserStudy>

7. Ke, Y., Tang, X., Jing, F.: The design of high-level features for photo quality assessment. In: IEEE Conf. Comp. Vis. Pat. Recog., vol. 1 (2006)
8. Luo, Y., Tang, X.: Photo and video quality evaluation: Focusing on the subject. In: Eur. Conf. Comp. Vis., pp. 386–399 (2008)
9. Obrador, P.: Region based image appeal metric for consumer photos. In: IEEE Work. Mult. Sig. Proc., pp. 696–701 (2008)
10. Tong, H., Li, M., Zhang, H., He, J., Zhang, C.: Classification of digital photos taken by photographers or home users. LNCS, pp. 198–205. Springer, Heidelberg (2004)
11. Moorthy, A.K., Bovik, A.C.: Visual importance pooling for image quality assessment. IEEE Jnl. Sel. Top. Sig. Proc. 3(2), 193–201 (2009)
12. Savakis, A.E., Etz, S.P., Loui, A.C.: Evaluation of image appeal in consumer photography. In: SPIE Proc., Human Vis. Elec. Img., pp. 111–121 (2000)
13. Datta, R., Li, J., Wang, J.Z.: Algorithmic Inferencing of Aesthetics and Emotion in Natural Images: An Exposition. In: IEEE Intl. Conf. Image Proc., pp. 105–108 (2008)
14. Amatriain, X., Pujol, J.M., Oliver, N.: I Like It, I Like It Not. In: Houben, G.-J., McCalla, G., Pianesi, F., Zancanaro, M. (eds.) UMAP 2009. LNCS, vol. 5535, pp. 247–258. Springer, Heidelberg (2009)
15. Pinson, M.H., Wolf, S.: Comparing subjective video quality testing methodologies. In: Vis. Comm. and Imag., SPIE, vol. 5150, pp. 573–582 (2003)
16. Beerends, J.G., De Caluwe, F.: The influence of video quality on perceived audio quality and vice versa. Jnl. Aud. Engg. Soc. 47, 355–362 (1999)
17. van Dijk, A.M., Martens, J.B., Watson, A.B.: Quality assessment of coded images using numerical category scaling. In: SPIE Adv. Image Video Comm. Storage Tech., vol. 2451, pp. 90–101 (1995)
18. BT.: 500-11: Methodology for the subjective assessment of the quality of television pictures. In: International Telecommunication Union, Geneva, Switzerland (2002)
19. LIVE video quality assessment databases,  
<http://live.ece.utexas.edu/research/quality/>
20. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Tran. Image Proc. 13, 600–612 (2004)
21. Yan, W., Kankanhalli, M.: Detection and removal of lighting & shaking artifacts in home videos. In: Proc. ACM Conf. Mult., pp. 107–116 (2002)
22. Dai, Z., Wu, Y.: Where Are Focused Places of a Photo? In: Qiu, G., Leung, C., Xue, X.-Y., Laurini, R. (eds.) VISUAL 2007. LNCS, vol. 4781, pp. 73–83. Springer, Heidelberg (2007)
23. Hasler, D., Susstrunk, S., GmbH, L., Steinfurt, G.: Measuring colourfulness in natural images. In: SPIE/IS&T Hum. Vis. Elec. Img., vol. 5007, pp. 87–95 (2003)
24. Cohen-Or, D., Sorkine, O., Gal, R., Leyvand, T., Xu, Y.: Color harmonization. In: Proc. ACM SIGGRAPH, vol. 25, pp. 624–630 (2006)
25. Li, C., Chen, T.: Aesthetic Visual Quality Assessment of Paintings. IEEE Jnl. Sel. Top. Sig. Proc. 3, 236–252 (2009)
26. Wang, Z., Sheikh, H.R., Bovik, A.C.: No-reference perceptual quality assessment of JPEG compressed images. In: IEEE Intl. Conf. Image Proc., vol. 1, pp. 477–480 (2002)
27. Richardson, I.: H. 264 and MPEG-4 video compression: video coding for next-generation multimedia. John Wiley & Sons Inc, Chichester (2003)
28. Freeman, M.: The image. revised edn. William Collins Sons & Co. Ltd. (1990)
29. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Heidelberg (2000)
30. Chang, C., Lin, C.: LIBSVM: a library for support vector machines (2001),  
<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

# Object Recognition Using Junctions

Bo Wang<sup>1</sup>, Xiang Bai<sup>1</sup>, Xinggang Wang<sup>1</sup>, Wenyu Liu<sup>1</sup>, and Zhuowen Tu<sup>2</sup>

<sup>1</sup> Dept. of Electronics and Information Engineering,  
Huazhong University of Science and Technology, China  
{wangbo.yunze,wxg}@gmail.com, {xbai,liuwu}@hust.edu.cn  
<sup>2</sup> Lab of Neuro Imaging, University of California, Los Angeles  
ztu@loni.ucla.edu

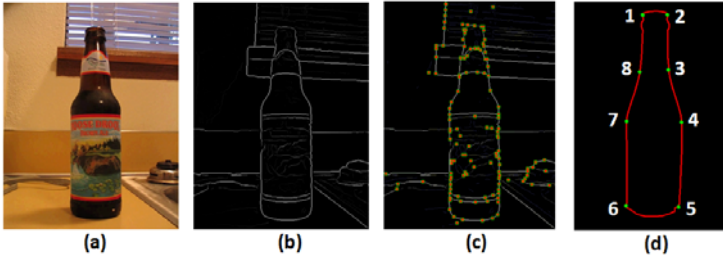
**Abstract.** In this paper, we propose an object detection/recognition algorithm based on a new set of shape-driven features and morphological operators. Each object class is modeled by the corner points (junctions) on its contour. We design two types of shape-context like features between the corner points, which are efficient to compute and effective in capturing the underlying shape deformation. In the testing stage, we use a recently proposed junction detection algorithm [1] to detect corner points/junctions on natural images. The detection and recognition of an object are then done by matching learned shape features to those in the input image with an efficient search strategy. The proposed system is robust to a certain degree of scale change and we obtained encouraging results on the ETHZ dataset. Our algorithm also has advantages of recognizing object parts and dealing with occlusions.

## 1 Introduction

Recent progress for object detection/recognition has been mostly driven by using advanced learning methods [2,3,4,5,6] and designing smart feature/object descriptors [7,8,9]. A detector is often trained on either a large number of features [2] or SIFT like features in a bounding box [4]. Most of the resulting algorithms, however, only tell whether an object is present or not in a bounding box by sweeping an input image at all locations and different scales. Besides the successes the field has witnessed for detecting rigid objects, such as frontal faces, detecting non-rigid objects remains a big challenge in computer vision and most of the systems are still not practical to use in general scenes [10].

Another interesting direction is using deformable templates [11] through matching-based approaches. Typical methods include generalized Hough transform [12], shape contexts [13], pyramid matching [14], pictorial structures [15], codebook-based approaches [16,17], and hierarchical shape representations [18,19,20]. These algorithms not only locate where an object appears in an image, they also recognize where the parts are, either through direct template correspondences or part representations. However, the performances of these algorithms are still not fully satisfactory, in terms of both efficiency and accuracy.

Marr [21] laid out a path to object recognition with a series of procedures including: (1) generic edge detection, (2) morphological operators such as edge



**Fig. 1.** (a) is the original image, (b) is an edge map by [11], (c) shows automatically detected junctions of (b), and (d) is our model with 8 junctions

linking and thinning, (3) shape matching on edges and object boundaries. This direction recently becomes unpopular because it largely relies on obtaining high quality edges; in addition, its object descriptors are too simplistic to handle the level of complexity in natural images. It is now accepted that perfect edge/feature detection does not exist [22] and it is hard to strictly separate the high-level recognition process from the low-level feature extraction stage. Nevertheless, these type of traditional methods still offer many appealing perspectives compared to modern approaches for being simple, generic, and without heavy learning.

In this paper, we take a rather traditional route by performing junction extraction first, followed by shape matching using a new set of feature descriptors. Note that from the remainder of this paper, we refer to junctions as corner points with more than one-degree connection. Given an object template described by its boundary contour, we annotate several corner points of high curvature with their order information; we then design two types of shape-context like features for describing the junction points. Note that these features are different from the standard shape context [13] since we only take into account the relevant junctions on the boundary. This means that, in the detection stage, we need to perform explicit search to exclude the background clutter. Fig. (II) shows an example of an object template with its corresponding junction points. To detect/recognize an object, we first apply a recently developed algorithm [1] to extract junction points from cluttered images; we then apply a pre-processing procedure to clean the edges and junctions; shape matching is then performed between the templates and the extracted junctions with an efficient search strategy. The proposed system spends about 1 or 2 minutes on an image to recognize an object (excluding another 1 or 2 minutes for extracting junctions). Our algorithm also has advantages of recognizing object parts and dealing with occlusions. The strength of this paper lies in: (1) the design of a new set of shape descriptors, (2) the development of a promising matching-based object detection/recognition system, (3) the achievement of significantly improved results on non-rigid objects like those in the ETHZ dataset.

There are several other related methods worth mentioning. Shotton et al. [23] describes the shape of the entire object using deformable contour fragments and their relative positions. Since their distance measure using improved Chamfer Matching is sensitive to the noise, many training samples are required for boosting the discriminative shape features. G. Heitz et al. [24] uses probabilistic shape to localize the object outlines. Our method is different from [25]. (1) We design two types of SC-like features of junctions and edges on actively searched contours whereas [25] uses geometric features of connected contours; (2) we emphasize a sparse representation on junctions whereas [25] uses dense points for object detection. Other works [26,27,18,28] decompose a given contour of a model shape into a group of contour parts, and match the resulting contour parts to edge segments in a given edge image.

## 2 Junction Features

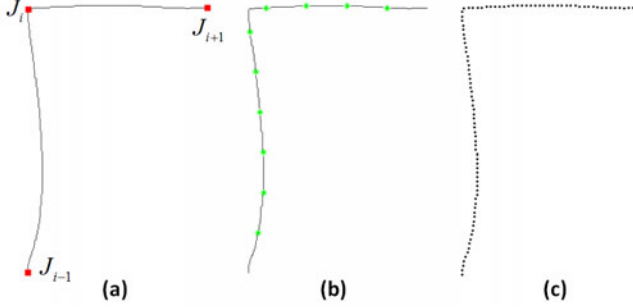
We use junction points as the basic elements to describe the object contour, and thus, our shape model could be considered as a simplified polygon with junction points being the vertices. In general, a majority of the junctions are the high curvature corner points of degree 2. There are also some junction points with degree 3 or 4, depending upon the image complexity. However, there are rarely junctions with degree higher than 4. We adopt a recently developed algorithm [1] to detect the junction points, and Fig. (1c) shows an example. In Fig. (1d), an object template with 8 junction points is displayed. As we can see, due to the presence of image clutter, it is not an easy task to match the template to the object even with reliable low-level features.

Given a contour  $C$  of  $n$  junction points, we denote  $C = (J_1, J_2, \dots, J_n)$ , where  $J_i$  is the  $i^{\text{th}}$  junction on  $C$ . Note that we preserve the clockwise order of each junction as the ordering is important in our model. In our current implementation, we assume multiple templates for each object type have the same number of junctions. However, clustering can be used to obtain different clusters of the same object type.

### 2.1 Junction Descriptors

We design two types of features, which are called  $F_1$  and  $F_2$  respectively. For each junction point  $J_i$ , we compute the feature  $F_1(J_i)$  based on its connected contour segments using a shape context like approach. Unlike the traditional shape context approaches [13] where all the points within the context radius are taken into account, we only use those points on the contour segments.

The two contour segments  $e_{i-1,i}$  and  $e_{i,i+1}$  between  $(J_{i-1}$  and  $J_i)$  and  $(J_i$  and  $J_{i+1})$  respectively are called **path** to  $J_i$ , denoted as  $P(J_i)$ . We then use path  $P(J_i)$  to characterize  $J_i$  and compute the corresponding feature  $F_1(J_i)$ . Fig. (2a) gives an example. We sample 10 points at equal space on  $P(J_i)$  and call them path points as  $(p_1^{(i)} \dots p_{10}^{(i)})$  (see green points in Fig. (2b)). Here  $t$  is the index along the path from  $J_{i-1}$  to  $J_{i+1}$ . Note that these 10 points are on the



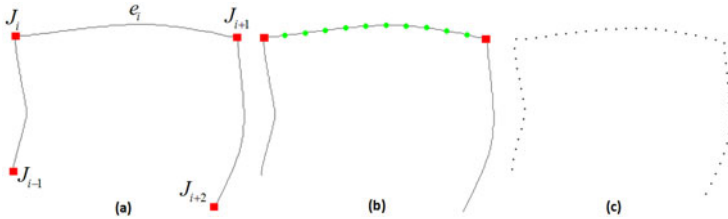
**Fig. 2.** The illustration for the feature ( $F_1$ ) for characterizing the junction points. The red ones in (a) are the junction points. The green dots in (b) are sampled path points on which shape-context like features are computed. (c) shows the densely sampled points for the green dots in (b) to compute shape context information.

path altogether and  $e_{i-1,i}$  and  $e_{i,i+1}$  may not have 5 points each since they do not necessarily have the same length. For each path point  $p_t$ , we compute its feature  $h(p_t)$  based on 50 densely sampled points on path  $P(J_i)$  at equal space. Fig. (2.c) gives an illustration. The parameter setting for computing the histogram of shape context is the same as that in [13]: 5 distance scales and 12 angle scales. Thus, each  $h(p_t)$  can be viewed as a feature vector of length 60. Finally, we are ready to describe  $F_1(J_i)$  as:

$$F_1(J_i) = (h(p_1^{(i)}), \dots, h(p_{10}^{(i)}))^T, \tag{1}$$

which is of length  $60 \times 10 = 600$ .

Next, we show how to compute feature  $F_2$  to characterize the shape information about a contour segment  $e_{i,i+1}$ . The approach is similar to the way  $F_1$



**Fig. 3.** The illustration for the feature ( $F_2$ ) for characterizing the contour segments between junction points. The red ones in (a) are the junction points. The green dots in (b) are those on which shape-context like features are computed. (c) shows the densely sampled points for the green dots in (b) to compute shape context information.

is computed. We sample 10 segment points at equal space on  $e_{i,i+1}$  and denote them as  $(p_1^{(i,i+1)} \dots p_{10}^{(i,i+1)})$ ; for each  $p_t^{(i,i+1)}$ , we compute its shape context feature based on 50 equally sampled points on  $e_{i-1,i}$ ,  $e_{i,i+1}$ , and  $e_{i+1,i+2}$  altogether; the parameter setting for computing the shape context is the same as that in computing  $F_1$ . This means that the features for  $e_{i,i+1}$  also takes into account its immediate neighboring segments. Thus,

$$F_2(e_{i,i+1}) = (h(p_1^{(i,i+1)}), \dots, h(p_{10}^{(i,i+1)}))^T, \quad (2)$$

which is also of length  $60 \times 10 = 600$ . Fig. (3) shows an illustration.

## 2.2 Junction Descriptors for Edge Maps

Due to the background clutter in natural images, the low-level edge/junction detection algorithms are always not perfect. We briefly describe some pre-processing steps in our algorithm. First, standard edge linking methods [29] are applied on extracted edge maps [1] using morphological operators. Fig. (4) gives an illustration. Fig. (4a) shows the original edge segments by [29], which removes many background clutters. The remaining edges are used to connect the junction points also by [1].



Fig. 4. The linking process for the segments around a junction

Given an input image  $I$  in the detection stage, we use method in [1] to extract the edges and junction points, and apply a software package [29] to perform edge linking. We call a junction point detected in a test image,  $J'$ . Next, we discuss how to compute its corresponding feature,  $F_1(J')$ . The idea is to search for the other two most plausible junctions  $J'_-$  and  $J'_+$  for  $J'$  to be adjacent on the object contour. The junctions on the template are selected based on the guideline to have high curvature; the search strategy echoes this but without using any shape matching strategy at this stage.

We first discuss the case where the degree of  $J'$  is 2. The problem is that the nearest junctions to  $J'$ ,  $J'_-(0)$  and  $J'_+(0)$  from the low-level edge/junction extraction process, might not be the desirable ones. We propose a simple deterministic procedure to perform the search:

- 1) Given a junction  $J'$ , we find its nearest junctions  $J'_-(0)$ ,  $J'_+(0)$  along the edges.



2) Let  $S = \{s_1, s_2, \dots, s_{|S|}\}$  denote the adjacent junctions of  $J'_-(t)$  on the edge map (the junctions on the path between  $J'$  and  $J'_-(t)$  are not included in  $S$ ).

Let  $\mathbf{x}$  and  $\mathbf{x}_-(t)$  be the coordinates of  $J'$  and  $J'_-(t)$  respectively. Let  $\mathbf{x}_l$  denote the coordinates of  $s_l$ . We compute the angle as:

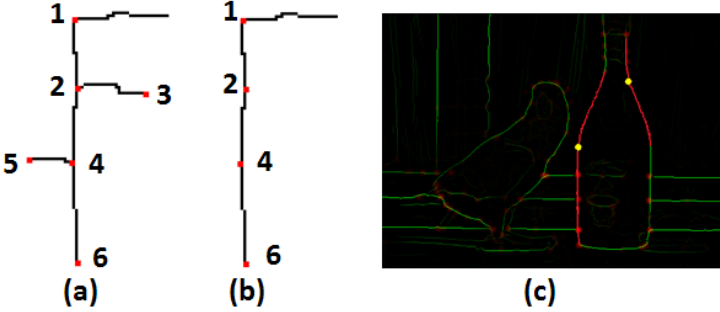
$$\theta_l = \arccos\left(\frac{(\mathbf{x} - \mathbf{x}_-(t)) \cdot (\mathbf{x}_-(t) - \mathbf{x}_l)}{|\mathbf{x} - \mathbf{x}_-(t)| |\mathbf{x}_-(t) - \mathbf{x}_l|}\right). \quad (3)$$

3) Then we find a  $l^*$  that satisfy:

$$l^* = \arg \min_{l=1,2,\dots,|S|} \theta_l. \quad (4)$$

If  $\theta_{l^*}$  is smaller than a given threshold  $\xi = 0.175$ , let  $t = t + 1$ , set  $s_{l^*}$  as  $J'_-(t)$  and go back to step 2). Else, output the  $J'_-(t)$  as the final  $J'_-$ .

The above procedures determine the junction  $J'_-$ , and the procedures to determine  $J'_+$  are the same. Fig. (5) shows an example when the degree of junction is 2. In Fig. (5)(a), point  $J'_1$  is a junction and the proposed procedure searches for the most plausible  $J'_-/J'_+$ , then the path between point 6 ( $J_-$ ) and point 1 ( $J'_+$ ) is chosen for computing feature  $F_1$ . Fig. (5)(b) shows the path between 1 and 6.



**Fig. 5.** An illustration for finding the path from junction  $J'_1$  (point 1) to its  $J'_-$  (point 6)

Once  $J'_-$  and  $J'_+$  are determined, we then obtain a path  $P(J')$  from  $J'_-$  to  $J'_+$  (passing  $J'$ ) that is used for computing the feature  $F_1(J')$ . Fig. (5)(c) gives two examples on a real image: the two points in yellow are two junctions, and the red segments denotes the paths used for computing  $F_1$  of them separately. We can also view our algorithm as designed for finding the salient contour segments, which might be useful in other vision applications.

When the degree of  $J'$  is higher than 2, we then compute multiple features  $F_1$  for  $J'$  corresponding different possible paths passing through  $J'$ . Let  $d(J')$  denote the degree of  $J'$ . There are  $d(J')$  junctions that are adjacent to  $J'$ , which

means there are  $D = C_{d(J')}^2$  paths  $P_q$  ( $q = 1, \dots, D$ ) that will pass  $J'$  ( $D$  pairs of  $J_-/J_+$  can be estimated). In order to keep the same with the 2-degree case, we consider  $J'$  as  $D$  different 2-degree junctions  $J'_{(q)}$  ( $q = 1, \dots, D$ ) with the same position and different paths (different  $F_1$  feature):

$$F_1(J'_{(q)}) = F_1(P_q). \quad (5)$$

Fig. 6 shows an example when the degree of a junction is 3. Point 1 in Fig. 6(a) can have three possible paths as separately shown in Fig. 6(b,c,d).

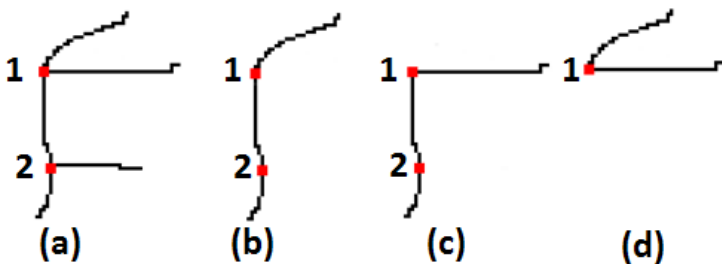


Fig. 6. An illustration for junctions with degree higher than 2

### 3 Detection

Once the junctions are determined, we then proceed to the detection/recognition stage by matching the features on the junctions and segments to those in the templates. A two-layer detection framework is proposed: In the first layer, we classify all the junctions in a edge map  $\mathcal{M}$  using a kNN classifier; based on the junction classification results, we use the shortest path to find the order of these junctions along the contour on  $\mathcal{M}$  in the second layer; then we localize the object position. Our goal is to find a sequence of junctions most similar to the training sequence, which is similar to shape matching with Dynamic Programming.

#### 3.1 Junction Classification

Recall that for each object type, all templates have the same number of junction points. For example, for the bottle templates, there are 8 junctions. Given a set of training templates, we compute the corresponding  $F_1(J)$  for each junction. The problem of computing how likely a junction  $J'$  in a test image belongs to a specific junction on the bottle becomes a classification problem. Each  $F_1(J)$  is a 600 dimension feature and we simply learn a kNN classifier to classify  $J'$  into  $\{1, 2, \dots, 8\}$  classes of junction points. In training a kNN classifier, the most important thing is to define the distance measure:

Let  $f$  denote a vector value of  $F_1$ , then we define the distance function  $dis$  in the same manner of SC [13]:

$$dis(F_1(J), F_1(J')) = \frac{1}{2} \sum_{i=1}^{600} \frac{(f_i - f'_i)^2}{f_i + f'_i}. \quad (6)$$

The class label  $L^*$  corresponding to the maximum is output by the algorithm:

$$L^* = \arg \max_{i=1, \dots, n} p(L_i | F_1(J')) \quad (7)$$

### 3.2 Graph Model

On the edge/junction map  $\mathcal{M}(I)$  of image  $I$ , we classify all the junctions into  $n$  groups  $G'_i$ , based on the trained kNN classifier. Our next goal is to localize the object boundary using a polygon with junctions as the vertices, which can be solved by finding the shortest path on a graph. As shown in Fig. (8), we construct a connected graph model  $(V, E)$  in which the vertices  $V$  represent junctions in a test image. Let  $e_{(j,k)}$  denote the edge between two junction nodes  $J'_j, J'_k$  from adjacent groups  $G'_i$  and  $G'_{i+1}$  respectively. Let  $w_{j,k}$  denote the weight of the edge  $e_{(j,k)}$ . We set two dummy node  $N_s$  and  $N_e$  (in red) as the source node and the target node respectively. The weights of the edges connecting with the two dummy points are set as zero. The intuition is that all the critical junctions on the object should lie on the shortest path between  $N_s$  and  $N_e$ . We use the shortest path algorithm to solve this problem.

The edge weight  $w_{j,k}$  is computed with dissimilarity between the edge  $e_{(j,k)}$  and the edges  $e_{i,i+1}^t (t = 1, \dots, M)$  from the training templates. We use  $F_2$  feature to measure this dissimilarity:

$$w_{j,k} = \frac{1}{M} \sum_{t=1}^M dis(F_2(e_{(j,k)}), F_2(e_{i,i+1}^t)) \quad (8)$$

Notice that the way for computing  $F_2$  feature on a edge map is different from the case for training template, since we do not know the adjacent junctions on a edge map. For junctions  $J'_j$  and  $J'_k$ , we can obtain their related paths  $P(J'_j)$  and  $P(J'_k)$  (as shown Fig. 7 (a)) respectively using the search algorithm proposed in Section 2.2 firstly. Then we sample the straight segment between  $J'_j$  and  $J'_k$  into ten points  $p_t^{(j,k)} (t = 1, \dots, 10)$  at equal space (see Fig. 7 (b)); For each  $p_t^{(j,k)}$ , we compute its shape contexts feature on 50 equally sample points (see Fig. 7 (c)) on  $P(J'_j)$  and  $P(J'_k)$  together. Finally, the  $F_2(e_{(j,k)})$  is described as:

$$F_2(e_{(j,k)}) = (h(p_1^{(j,k)}), \dots, h(p_{10}^{(j,k)}))^T, \quad (9)$$

For the shortest path on our graph model, the linear programming has the special property that is integral. A \* search algorithm [19] which uses heuristics to try to speed up the search can be applied to solve the optimization problem. The confidence for a detection is the sum of all the edge weights on the shortest path, which is used for the categories classification.

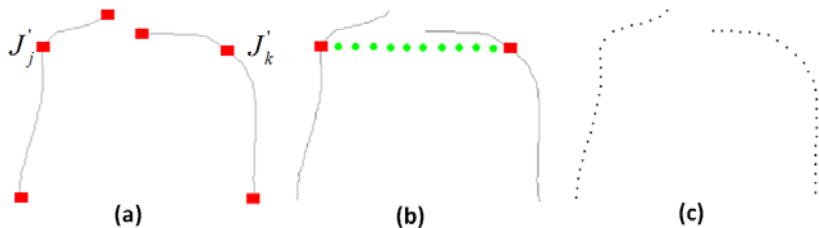


Fig. 7. The illustration for computing  $F_2$  feature on a edge map

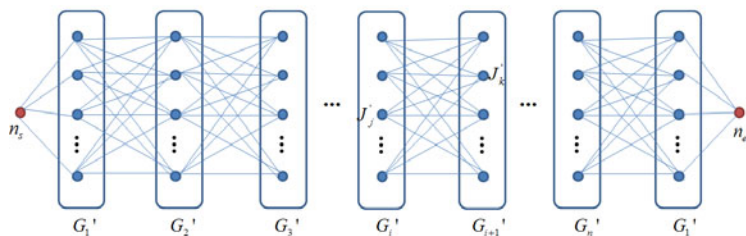


Fig. 8. The illustration for the graph model

## 4 Experiments

We tested the proposed method on ETHZ shape dataset [16], which contains 5 different shape-based classes (apple logos, bottles, giraffes, mugs, and swans) with 255 images in total. Each category has significant variations in scale, intra-class pose, and color which make the object detection task challenging. To have a fair comparison with [28], we use 1/3 positive images of each class as training samples, same to [28]. Fig. 9 shows a few training contour templates (the red

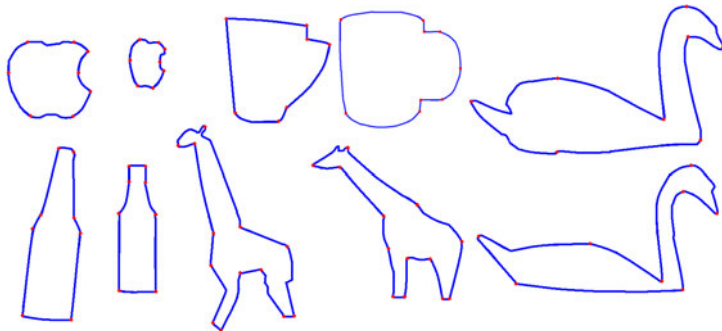
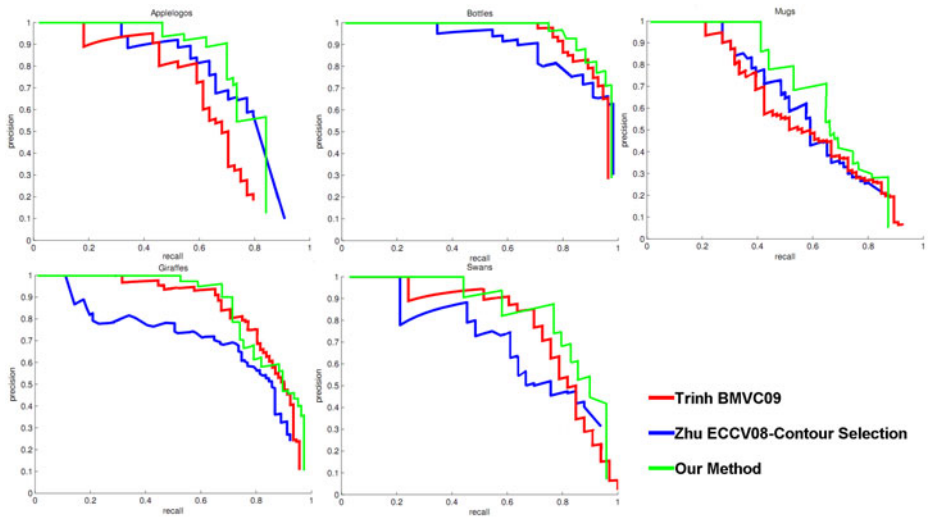


Fig. 9. Two training templates for each class from the ETHZ dataset [16]

points denote the junctions of each contour), which are extracted from the binary masks of the ETHZ dataset.

Initially, we extracted the gPb-based edge maps and junctions [1]. In an image of average complexity, there are on average 100 junctions. We take the binary mask annotation as the training templates. The results under PASCAL criterion of our method are reported in Fig. (10) with precision vs. recall curve. We also compare it to the latest results in [28,26]. Fig. (10) shows P/R curves for the Kimia’s method based on skeletal shape model [28] in red and for contour selection [26] in blue. Our method significantly outperforms [28] on the four categories of apple logos, bottles, mugs and swans, and a little better than [28] in the category of giraffe. This demonstrates that our junction model can well capture the intra-class variations of objects. Our result is also better than [26] on all the categories. We also compare the precision at the same recall to [28,26,30].

As Table 1 shows, our method works better than [28] and [30], particularly in the category of apple logos. This is because our junction features take into consideration of both local and global structures. Even though our method is



**Fig. 10.** Precision/Recall curves of our method compared to [28] and [26] for 5 classes of ETHZ dataset

**Table 1.** Comparison of the precision at the same recall

	Apple logos	Bottles	Giraffes	Mugs	Swans
Our method	<b>52.9/86.4</b>	<b>69.8/92.7</b>	<b>82.4/70.3</b>	<b>28.2/83.4</b>	<b>40.0/93.9</b>
Zhu et al. [26]	49.3/86.4	65.4/92.7	69.3/70.3	25.7/83.4	31.3/93.9
Trinh&Kimia [28]	18.0/86.4	65.1/92.7	80.0/70.3	26.3/83.4	26.3/93.9
Ferrari et al. [30]	20.4/86.4	30.2/92.7	39.2/70.3	22.7/83.4	27.1/93.9

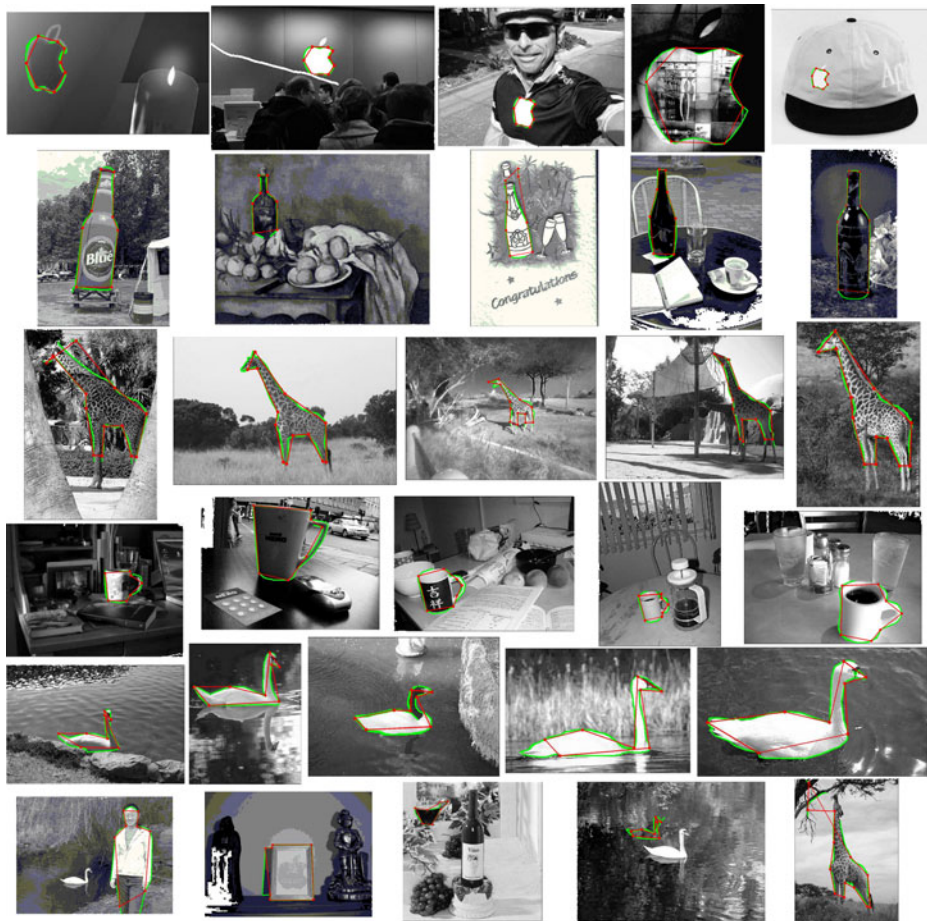
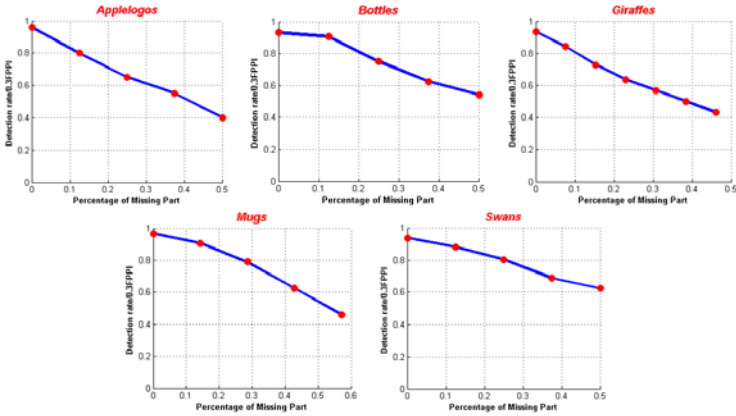


Fig. 11. The detection results of ETHZ dataset

just slightly better than [28] in the category of giraffe, our method does not need multi-scale shape skeletons as our method is based on junctions that are more-or-less scale-invariant. Fig. (11) shows some detection results by the proposed method. The points and segments in red are the junction points and the polygons that use the detection junctions as the vertices. We also show the contour segments (in green) related to each junction in these images; we observe that our method not only can detect the object position robustly but also have good localization of the object contour, benefiting from the junctions.

The last row in Fig. (11) shows a few false detections. It's very interesting that we detected a girl when detecting a bottle in the first image (last row); in the second image, we detected a photo frame when detecting a mug; in the third image, we detected a mug when detecting a swan; the fourth and fifth images



**Fig. 12.** The curves about detection rate (DR) at 0.3 FPPI vs. the percentage of miss contours for 5 classes of ETHZ dataset



**Fig. 13.** The detection results for partial contour detection with the proposed method

(last row) show two examples about false positives. Notice that even we could not detect a swan in the fourth image, the segments detected out are very similar to a swan, which is a graceful failure.

Our method is not limited to detect the whole contour of objects. It can also be used to detect object parts. For detecting a contour part, we only use a group of consecutive junctions from  $J_i^t$  to  $J_{i+m}^t$  ( $m < n$ ) on the training templates. We randomly choose the start junction and end junction with a fixed length percentage for training, and to make a clear evaluation of performance, we use detection rate vs false positive per image(DR/FPPI). Fig. 12 reports the average detection rate at 0.3 FPPI for five classes of ETHZ dataset. In Fig. 12, we observe that our detection rates can still reach above 0.4 at 0.3 FPPI when 50% of the training contours are missing. This demonstrates that the proposed junction

features are stable and effective for recognizing shapes in clutter images. Fig. 13 shows a few detection results with only parts detected.

## 5 Conclusions and Future Work

In this paper, we have introduced a shape-based object detection/recognition system and showed its advantage on detecting rigid and non-rigid objects, like those in the ETHZ dataset. Our method follows the line of template matching by defining contour templates with a set of junction points. We found the designed shape descriptors to be informative and our system outperforms many contemporary approaches using heavy learning and design. We anticipate junction features to be useful for other vision tasks. In the future, we plan to combine the shape features with appearance information to provide more robust results.

## Acknowledgement

We thank Michael Maire for nicely providing us the edge images with junctions of ETHz dataset. This work was jointly supported by NSFC 60903096, NSFC 60873127, ONR N000140910099, NSF CAREER award IIS-0844566, and China 863 2008AA01Z126.

## References

1. Maire, M., Arbelaez, P., Fowlkes, C., Malik, J.: Using contour to detect and localize junctions in natural images. In: CVPR (2008)
2. Viola, P.A., Jones, M.J.: Robust real-time face detection. IJCV 57, 137–154 (2004)
3. Kumar, S., Hebert, M.: Discriminative random fields: a discriminative framework for contextual interaction in classification. In: ICCV (2003)
4. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multi-scale, deformable part model. In: CVPR (2008)
5. Dollár, P., Babenko, B., Belongie, S., Perona, P., Tu, Z.: Multiple component learning for object detection. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 211–224. Springer, Heidelberg (2008)
6. Maji, S., Malik, J.: Object detection using a max-margin hough transform. In: CVPR (2009)
7. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
8. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. of CVPR, pp. 886–893 (2005)
9. Gu, C., Lim, J.J., Arbelaez, P., Malik, J.: Recognition using regions. In: CVPR (2009)
10. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: Proc. of CVPR (2008)
11. Yuille, A.L., Hallinan, P.W., Cohen, D.S.: Feature extraction from faces using deformable templates. IJCV 8, 99–111 (1992)
12. Ballard, D.H.: Generalizing the hough transform to detect arbitrary shapes. Pattern Recognition 13, 111–122 (1981)



13. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. *PAMI* 24, 509–522 (2002)
14. Grauman, K., Darrell, T.: The pyramid match kernel: Discriminative classification with sets of image features. In: *Proc. of ICCV* (2005)
15. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. *IJCV* 61, 55–79 (2005)
16. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of adjacent contour segments for object detection. *PAMI* 30, 36–51 (2008)
17. 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. 3952, pp. 575–588. Springer, Heidelberg (2006)
18. Gavrilu, D.: A bayesian, exemplar-based approach to hierarchical shape matching. *IEEE Trans. Pattern Anal. Mach. Intell.* 29, 1408–1421 (2007)
19. Felzenszwalb, P.F., Schwartz, J.D.: Hierarchical matching of deformable shapes. In: *CVPR* (2007)
20. Zhu, L., Chen, Y., Ye, X., Yuille, A.L.: Structure-perceptron learning of a hierarchical log-linear model. In: *Proc. of CVPR* (2008)
21. Marr, D.: *Vision*. W.H. Freeman and Co., San Francisco (1982)
22. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, colour and texture cues. *PAMI* 26, 530–549 (2004)
23. Shotton, J., Blake, A., Cipolla, R.: Multi-scale categorical object recognition using contour fragments. *PAMI* 30, 1270–1281 (2008)
24. Heitz, G., Elidan, G., Parker, B., Koller, D.: Loops: Localizing object outlines using probabilistic shape. In: *NIPS* (2008)
25. Ferrari, V., Jurie, F., Schmid, C.: Accurate object detection combining recognition and segmentation. In: *CVPR* (2007)
26. Zhu, Q., Wang, L., Wu, Y., Shi, J.: Contour context selection for object detection: A set-to-set contour matching approach. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part II*. LNCS, vol. 5303, pp. 774–787. Springer, Heidelberg (2008)
27. Lu, C., Latecki, L., Adluru, N., Yang, X., Ling, H.: Shape guided contour grouping with particle filters. In: *ICCV* (2009)
28. Trinh, N., Kimia, B.: Category-specific object recognition and segmentation using a skeletal shape model. In: *BMVC* (2009)
29. Kovesi, P.D.: *MATLAB and Octave functions for computer vision and image processing*. School of Computer Science & Software Engineering, The University of Western Australia (2008), <http://www.csse.uwa.edu.au/~pk/research/matlabfns/>
30. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: From images to shape models for object detection. Technical Report 6600, INRIA (2008)

# Using Partial Edge Contour Matches for Efficient Object Category Localization

Hayko Riemenschneider, Michael Donoser, and Horst Bischof\*

Institute for Computer Graphics and Vision,  
Graz University of Technology, Austria  
{hayko,donoser,bischof}@icg.tugraz.at

**Abstract.** We propose a method for object category localization by partially matching edge contours to a single shape prototype of the category. Previous work in this area either relies on piecewise contour approximations, requires meaningful supervised decompositions, or matches coarse shape-based descriptions at local interest points. Our method avoids error-prone pre-processing steps by using all obtained edges in a partial contour matching setting. The matched fragments are efficiently summarized and aggregated to form location hypotheses. The efficiency and accuracy of our edge fragment based voting step yields high quality hypotheses in low computation time. The experimental evaluation achieves excellent performance in the hypotheses voting stage and yields competitive results on challenging datasets like ETHZ and INRIA horses.

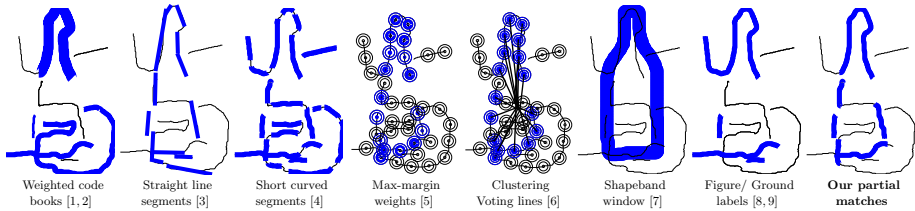
## 1 Introduction

Object detection is a challenging problem in computer vision. It allows localization of previously unseen objects in images. In general, two main paradigms can be distinguished: appearance and contour. Appearance-based approaches form the dominant paradigm using the bag-of-words model [10], which analyzes an orderless distribution of local image features and achieves impressive results mainly because of powerful local image description [11].

Recently, the contour-based paradigm has become popular, because shape provides a powerful and often more generic feature [12] since an object contour is invariant to extreme lighting conditions and large variations in texture or color. Many different contour-based approaches exist and the research falls mainly into four categories. The works of [11,2] focus on the aspect of learning edge codebooks, where chamfer matching is used to evaluate local shape similarity. Other research uses piecewise approximations of the edges by short segments [13,4] or supervised decompositions [8]. In [5,6,14] the problem is cast as a matching between shape-based descriptors on local interest points.

---

\* This work was supported by the Austrian Research Promotion Agency (FFG) project FIT-IT CityFit (815971/14472-GLE/ROD).



**Fig. 1.** Overview of related work: Our approach relaxes the piecewise approximations and local neighborhoods. We use partial matching to find contour fragments belonging to the foreground rather than discarding entire edges. See Section 2 for details.

The main motivation for our work is that *“connectedness is a fundamental powerful driving force underexploited in object detection”* [3]. Viewing edge contours as connected sequences of any length instead of short segment approximations or local patches on interest points provides more discrimination against background clutter. In our contributions we focus on the partial matching of noisy edges to relax the constraints on local neighborhoods or on assigning entire edges as background disregarding local similarities. We formulate a category localization method which efficiently retrieves partial edge fragments that are similar to a single contour prototype. We introduce a self-containing descriptor for edges which enables partial matching and an efficient selection and aggregation of partial matches to identify and merge similar overlapping contours up to any length. A key benefit is that the longer the matches are, the more they are able to discriminate between background clutter and the object instance. In this way we lift standard figure / ground assignment to another level by providing local similarities for all edges in an image. We retrieve these partial contours and combine them directly in a similarity tensor and together with a clustering-based center voting step we hypothesize object locations. This greatly reduces the search space to a handful of hypotheses and shows excellent performance compared to state of the art in the voting stage. For a full system evaluation, the hypotheses are further verified by a standard multi-scale histogram of gradients (HOG) classifier.

## 2 Related Work

There exists a range of work in the contour-based paradigm which achieve state-of-the-art performance for several object categories using contour information, for an overview see Figure 1. The research falls into four main categories, namely (i) learning codebooks of contour fragments, (ii) approximating contours by piecewise segments, (iii) using local description of the contour at selected interest points, or (iv) assigning entire edges to either foreground or background. Additional techniques are used in each work, for example learning deformation models, sophisticated cost functions or probabilistic grouping.

*Learning codebooks:* Shotton et al. [1] and Opelt et al. [2] concurrently proposed to construct shape fragments tailored to specific object classes. Both find matches to a pre-defined fragment codebook by chamfer matching to the query image and then find detections by a star-shaped voting model. Their methods rely on chamfer matching which is sensitive to clutter and rotation. In both approaches the major aspect is to learn discriminative combinations of boundary parts as weak classifiers using boosting to build a strong detector.

*Piecewise approximation:* Ferrari et al. [15,3] build groups of approximately straight adjacent segments (kAS) to work together in a team to match the model parts. The segments are matched within a contour segmentation network which provides the combinations of multiple simple segments using the power of connectedness. In later work they also show how to automatically learn codebooks [3], or how to learn category shape models from cropped training images [16]. In a verification step they use a thin-plate-spline (TPS)-based matching to accurately localize the object boundary. Similar to this, Ravishankar et al. [4] use short segments to approximate the outer contour of objects. In contrast to straight segments, they prefer slightly curved segments to have better discriminative power between the segments. They further use a sophisticated scoring function which takes local deformations in scale and orientations into account. However, they break the reference template at high curvature points to be able to match parts, again resulting in disjoint approximations of the actual contour. In their verification stage, the gradient maps are used as underlying basis for object detection avoiding the error-prone detection of edges.

*Shape-based interest points:* This category uses descriptors to capture and match coarse descriptions of the local shape around interest points. Leordeanu et al. [17] use simple features based on normal orientations and pairwise interactions between them to learn and detect object models in images. Their simple features are represented in pairwise relations in category specific models that can learn hundreds of parts. Berg et al. [14] formulate the object detection problem as a deformable shape matching problem. However, they require hand-segmented training images and do not learn deformation models in training. Further in the line are the works of Maji and Malik [5] and Ommer and Malik [6] which match geometric blur features to training images. The former use a max-margin framework to learn discriminative weights for each feature type to ensure maximal discrimination during the voting stage. The latter provide an interesting adaptation of the usual Hough-style center voting. Ommer and Malik transform the discrete scale voting to a continuous domain where the scale is another unknown in the voting space. Instead of multiple discrete center vectors, they formulate the votes as lines and cluster these to find scale-coherent hypotheses. The verification is done using a HOG-based fast SVM kernel (IKSVM).

*Figure / ground assignment:* Similar in concept but not in practice are the works of Zhu et al. [8] and Lu et al. [9]. They cast the problem as figure / ground labeling of edges and decide for a rather small set of edges which belong to

the foreground and which are background clutter. By this labeling they reduce the clutter and focus on salient edges in their verification step. Lu et al. use particle filters under static observation to simultaneously group and label the edge contours. They use a new shape descriptor based on angles to decide edge contour similarity. Zhu et al. use control points along the reference contour to find possible edge contour combinations and then solve cost functions efficiently using linear programming. They find a maximal matching between a set of query image contours and a set of salient contour parts from the reference template, which was manually split into a set of reference segments. Both assume to match entire edge contours to the reference sets and require long salient contours. Recent work by Bai et al. [7] is also based on a background clutter removal stage called shapeband. Shapeband is a new type of sliding window adapted to the shape of objects. It is used to provide location hypotheses and to select edge contour candidates. However, in their runtime intensive verification step they iteratively compute shape context descriptors [18] to select similar edge contours. Another recent approach by Gu et al. [19] proposes to use regions instead of local interest points or contours to better estimate the location and scale of objects.

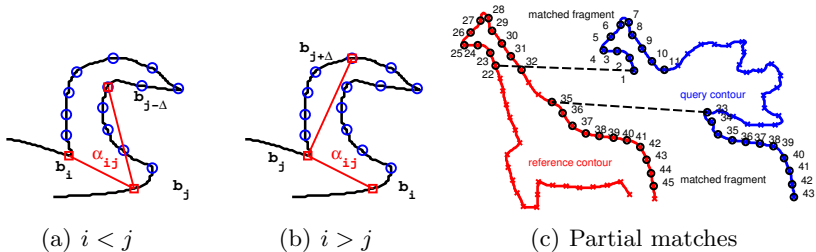
We place our method in between the aforementioned approaches. We use edge contours in the query image and match them at any length from short contour segments up to full regions boundaries using partial shape matching. In such a setting the similarity to the prototype shape decides the complexity and length of the considered contours.

### 3 Partial Shape Matching for Object Detection

In the following sections we describe our proposed approach to detect objects by computing partial similarities between edge contours in a query image and a reference template. For the sake of clarity, we will now define some terms used throughout the paper, see Figure 2 for a visual illustration. We use the term fragment to denote a part of an edge contour. Edge contours can be arbitrarily long, contain irrelevant parts or may also be incomplete due to missing edge detector responses or occlusions which make parts of the object invisible. The query contours are the connected edge contours found by the detector and subsequent 8-neighborhood linking. The reference contour is a single hand-drawn model of the object’s outer boundary. A valid matched fragment is defined as a part of an edge contour that is similar to a part of the reference contour.

Our goal is to identify matches from fragments of arbitrary length (contained within the query edges) to the reference contour, by analyzing a self-contained representation and description of the shape of the detected edge contours. We want to build a representation that contains the whole as well as any part of a contour, which enables matching independently from the remaining parts.

Our detection method consists of three parts: First, edges are extracted from an image and represented as lists of coordinates. This representation is the basis for the self-containing contour descriptor. Second, the matching is a vital component which allows the efficient retrieval of contour fragments similar to a given



**Fig. 2.** Illustration of 2D angle description and matching. a-b) An angle is measured between any two sampled points  $b_i$  and  $b_j$ , which define a fragment inside an edge contours, c) shows partial matches in an occluded edge to a reference contour.

reference prototype. Third, for each matched fragment we calculate a center vote to estimate the location of the searched object and aggregate coherent fragments based on their voting, scale and correspondence to the reference.

### 3.1 Fragment Description

Our goal is to exploit the connectedness of an edge contour implicitly yet allowing to retrieve parts of an edge as fragments. Many different methods have been proposed for partial contour matching. Angular representations are a natural choice due their direct encoding of geometric layout. For example, Turney et al. [20] use the slope  $\theta$  and arc length  $s$  as local representation for boundaries, however only on a small set of images. In a more recent work Brendel and Todorovic [21] find matching fragments in complex images using circular dynamic time warping with a runtime of 200ms per match. Chen et al. [22] proposed an efficient matching, however their descriptor only measures local shape and ignores global contour similarity. Felzenszwalb et al. [23] proposed a hierarchy of deformable shapes where only a single contour can be matched in subtrees and matching two contours requires 500ms. A recent hierarchical approach by Kokkinos and Yuille [24] formulates the task as image parsing and provide fast coarse to fine matches. Lu et al. [9] developed a shape descriptor based on a 3D histogram of angles and distances for triangles connecting points sampled along the contours. They do not allow partial matching and the descriptor requires high computational costs. Donoser et al. [25] developed a descriptor which can be seen as a subset of [9], where angles between any two points and a fixed third point on a closed contour are analyzed. They demonstrate efficient matching between two closed shapes within a few milliseconds.

Inspired by the high quality of hierarchical approaches, we adopt the descriptor from Donoser et al. [25], which was designed for matching whole object silhouettes, to handle the requirements of object detection in cluttered images. First, partial matching of the cluttered edges must be possible. Since there are no closed contours around a cluttered object after standard edge detection, we

design a novel self-containing descriptor which enables efficient partial matching. Second, similar to hierarchies the descriptor encodes coarse and fine contour information. Different sampling of the descriptor enables direct access to different levels of detail for the contour, whereas the full descriptor implicitly contains all global and local contour information.

The method in [25] proposed an efficient matching step to describe and then retrieve all redundant and overlapping matching combinations. Their brute force algorithm delivers good results on clean silhouette datasets. However, for an object detection task this is not feasible due to the prohibitive combinatorics (multiple scales, multiple occlusions and hundreds of edges per image). Additionally, slightly shifted matches at neighboring locations contradict each other and do not provide coherent object location hypotheses. Therefore, in contrast to [25], we propose an efficient summarization scheme directly in an obtained 3D similarity tensor. Such an approach has several strong benefits like selection and aggregation of only coherent center vote matches, longer merged matches out of indiscriminate shorter segments and further an immense speedup due to the reduction of the number of returned matches. The main motivation is to exploit the connectedness of edge contours instead of using individual interest points or short piecewise approximations of edge contours.

As a first step we sample a fixed number  $N$  of points from the closed reference contour that can be ordered as  $R = \{r_1, r_2, \dots, r_N\}$ . As next step we have to extract connected and labeled edge contours from the query image. Edge detection and linking in general is a quite challenging task [26]. We apply the Pb edge detector [27] and link the results to a set of coordinate lists. For the obtained query contours, points are sampled at equal distance, resulting in a sequence of points  $B = \{b_1, b_2, \dots, b_M\}$  per contour. The sampling distance  $d$  between the points allows to handle different scales. Sampling with a larger distance equals to a larger scale factor, and vice versa. For detecting objects in query images we perform an exhaustive search over a range of scales, which is efficiently possible due to the properties of our descriptor and matching method.

We use a matrix of angles which encode the geometry of the sampled points leading to a translation and rotation invariant description for a query contour. The descriptor is calculated from the relative spatial orientations between lines connecting the sampled points. In contrast to other work [9,25], we calculate angles  $\alpha_{ij}$  between a line connecting the points  $b_i$  and  $b_j$  and a line to a third point relative to the position of the previous two points. This angle is defined

$$\alpha_{ij} = \begin{cases} \sphericalangle(\overline{b_i b_j}, \overline{b_j b_{j-\Delta}}) & \text{if } i < j \\ \sphericalangle(\overline{b_i b_j}, \overline{b_j b_{j+\Delta}}) & \text{if } i > j \\ 0 & \text{if } \text{abs}(i - j) \leq \Delta \end{cases}, \quad (1)$$

where  $b_i$  and  $b_j$  are the  $i^{\text{th}}$  and  $j^{\text{th}}$  points in the sequence of sampled points of the contour and  $\Delta$  is an offset parameter of the descriptor (5 for all experiments). See Figure 2 for an illustration of the choice of points along the contour. The third point is chosen depending on the position of the other two points to ensure

that the selected point is always inside the contour. This allows us to formulate the descriptor as a self-containing descriptor of any of its parts.

The angles  $\alpha_{ij}$  are calculated between every pair of points along a contour. In such a way a contour defined by a sequence of  $M$  points is described by an  $M \times M$  matrix where an entry in row  $i$  and column  $j$  yields the angle  $\alpha_{ij}$ . Figure 3 illustrates the descriptors for different shape primitives. The proposed descriptor has four important properties. First, its angular description makes it translation and rotation invariant. Second, a shift along the diagonal of the descriptor handles the uncertainty of the starting point in edge detection. Third, it represents the connectedness of contours by using the sequence information providing a local (close to matrix diagonal) and global (far from matrix diagonal) description. And most importantly, the definition as a self-containing descriptor allows to implicitly retrieve partial matches which is a key requirement for cluttered and broken edge results.



**Fig. 3.** Visualization of descriptors for selected contour primitives. Middle row shows descriptors from [25] and bottom row shows our descriptors. Note how each fragment is included in its respective closed contours (square and circle) in our version, which is not fulfilled for [25] since it was designed for closed contour matching.

### 3.2 Fragment Matching and Merging

Matching and merging partial contours is an important part of our approach and is based on the 2D edge contour descriptors introduced in the previous section. For any two descriptors representing two contours, the aim of matching is to identify parts of the two contours which are similar to each other. In terms of comparing corresponding descriptor matrices, one has to compare all sub-blocks of the descriptor matrices to find all matching possibilities and lengths. For efficient calculation of all similarity scores, we apply the algorithmic optimization using integral images as proposed in [25] to access the partial descriptor differences in constant time, which returns the similarities (differences between our angle descriptors) for all matching triplets  $\{r, q, l\}$  stored in a 3D similarity and correspondence tensor  $\Gamma_{(r,q,l)}$ . The first two dimensions identify the starting points of the match in the reference (r) and query edge contour (q) and the third



dimension defines the length ( $l$ ) of the match. Note that this tensor fully defines all possible correspondences between the reference and the edge contour. Figure 4 shows the similarity tensor for the partial matching example in Figure 2. The two matched fragments correspond to the peaks (a) in the tensor, here a single slice at a fixed length  $l = 11$  is shown.

A main issue is redundancy within the tensor as there may be many overlapping and repetitive matches. This poses a problem for object detection in cluttered images. Our goal is to find the longest and most similar fragments and merge repetitive matches instead of retrieving all individual matches. This is an important part of this work. First, it is necessary to outline some of the properties of our 3D similarity and correspondence tensor  $\Gamma_{(r,q,l)}$ .

- I. A fragment  $(r, q, l)$  is assigned a similarity (Euclidean distance between angular descriptors) by  $\Gamma_{(r,q,l)}$ .
- II. Length variations  $(r, q, l_2)$  with  $l_2 < l$  define the same correspondence, yet shorter in length.
- III. Diagonal shifts in the indices  $(r + 1, q + 1, l)$  also represent the same match, yet one starting point *later*.
- IV. Unequal shifts  $(r + 1, q, l)$  define a different correspondence, however very similar and close.
- V. Due to occlusions or noise, multiple matches per edge contour may exist. The example in Figure 2 is a shifted match *much later*  $(r + 13, q + 32, l)$  defining the same correspondence, yet skipping  $(32-13=19)$  points of noise.
- VI. Matches near to the end of each contour (if not closed) have a maximal length given by the remaining points in each contour sequence.

Perfect matches would result in singular *peaks* in a slice. However due to these small shifts along the same correspondence or with an unequal offset, matches result in a *hill*-like appearance of the similarity, see Figure 4a. Given these properties we now define a matching criterion to deliver the longest and most similar matches, i.e. finding the peaks not once per slice but for the entire 3D tensor. This summarization is made of three steps: (a) finding valid correspondences satisfying the constraints on length and similarity, (b) merging all valid correspondences to obtain the longest combination of the included matches (property II) and (c) selecting the maximal similarity of matches in close proximity (property IV). The steps are in detail as following:

First, we define a function  $\mathcal{L}(r, q, l)$  which gives the lengths at any given valid correspondence tuple  $r, q$  as

$$\mathcal{L}(r, q, l) = \begin{cases} l & \text{if } \Gamma_{(r,q,l)} \leq s_{lim} \text{ and } l \geq l_{lim} \\ 0 & \text{else} \end{cases}, \quad (2)$$

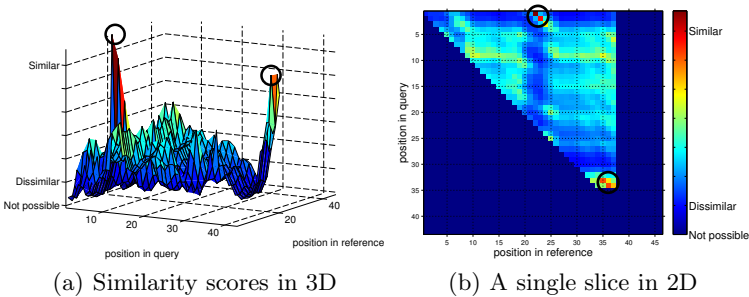
where the value at  $\mathcal{L}(r, q, l)$  is the length of a valid fragment. A valid fragment has a similarity score below the limit  $s_{lim}$  and a minimal length limit of  $l_{lim}$ . This function is used to define a subset of longest candidates by

$$\Psi_{(r,q)} = \forall_{r,q} : \arg \max_{l \in \min(N,M)} \mathcal{L}(r, q, l), \quad (3)$$

where  $\Psi_{(r,q)}$  is a subset of  $\Gamma_{(r,q,l)}$  containing the longest matches at each correspondence tuple  $(r,q)$ . This set contains matches for every possible correspondence given by the constraints on similarity and matching positions (see property II, VI). However, we further want to reduce this to only the local maxima (conserving property IV). Since the set can now be considered as a 2D function, we find the connected components  $\mathcal{C}$  satisfying  $\Psi_{(r,q)} > 0$ . The final set of candidates are the maxima per connected component and is defined as

$$\Upsilon_{(r,q,l)} = \forall c_i \in \mathcal{C} : \arg \max_{\Gamma_{(r,q,l)}} (\Psi_{(r,q)} \in c_i), \quad (4)$$

where  $\Upsilon_{(r,q,l)}$  holds the longest possible and most similar matches given the constraints on minimum similarity  $s_{lim}$  and minimal length  $l_{lim}$ . In the example shown in Figure 2 and 4 the final set contains two matches, which are the longest possible matches. Note that shorter matches in the head and back are possible, but are directly merged to longer and more discriminative matches by analyzing the whole tensor. Furthermore, obtained matches are local maxima concerning similarity scores. This provides an elegant and efficient summarization leading to coherent and discriminative matches and reduced runtime.



**Fig. 4.** Illustration of the similarity and correspondence tensor  $\Gamma_{(r,q,l)}$  at length  $l = 11$  for example shown in Figure 2(c): (a) the two peaks correspond to the matches found. Matching uncertainty results in multiple peaks in a *hill*-like appearance. (b) shows the same similarity in a flat view, where red signals high similarity and dark blue defines invalid matches due to length constraints. Best viewed in color.

### 3.3 Hypothesis Voting

Matching as described in the previous section provides a set of matched fragments for the query edges, which have to be combined to form object location

hypotheses. In the following we describe how matched fragments are grouped for object locations hypotheses and scores are estimated.

**Fragment Aggregation.** Up to this point we have a set  $\mathcal{Y}_{(r,q,l)}$  of matched parts of edge contours detected in a query image which are highly similar to the provided prototype contour. Every match has a certain similarity and length. Further, we can map each matched contour to its reference contour and estimate the object centroid from the given correspondence tuple. The aggregation of the individual fragments identifies groups of fragments which complement each other and form object location hypotheses.

For this step we cluster the matched fragments analyzing their corresponding center votes and their scale by mean-shift mode detection with a scale-dependent bandwidth. The bandwidth resembles an analogy to the classical Hough accumulator bin size, however with the added effect that we combine the hypotheses locations in a continuous domain rather than discrete bins.

**Hypothesis Ranking.** All obtained hypotheses are ranked according to a confidence. For this purpose we investigate two ranking methods. The first is based on the coverage of detected fragments, where  $\zeta_{COV}$  is a score relative to the amount of the reference contour that is covered by the matched fragments, defined as

$$\zeta_{COV} := \frac{1}{N} \sum_{i=1}^N (f_i \times S_i), \quad (5)$$

where  $f_i$  is the number of times the  $i$ -th contour point has been matched and  $S_i$  is the corresponding weight of this point. This is normalized by the number of contour points  $N$  in the reference contour. The coverage score  $\zeta_{COV}$  provides a value describing how many parts are matched to the reference contour for the current hypothesis. We use a uniform weight of  $S_i = 1$ . However, for example weights given by the contour flexibility [28] would be an interesting aspect.

As a second score, we use a ranking as proposed by Ommer and Malik [6]. They define the ranking score  $\zeta_{PMK}$  by applying an SVM classifier to the image windows around the location hypotheses. The kernel is the pyramid match kernel (PMK) [29] using histograms of oriented gradients (HOG) as features. Positive samples for each class are taken from the ground truth training set. Negative samples are retrieved by evaluating the hypotheses voting and selecting the false positives. The bounding boxes are resized to a fixed height while keeping median aspect ratio. Since the mean-shift mode detection may not deliver the true object location, we sample locations in a grid of windows around the mean-shift center. At each location we evaluate the aforementioned classifier and retrieve the highest scoring hypothesis as new detection location.

## 4 Experiments

We demonstrate the performance of our proposed object category localization method on two different reference data sets: ETHZ (Section 4.1) and INRIA

**Table 1.** Hypothesis voting, ranking and verification stages show competitive detection rates using PASCAL criterion for the ETHZ shape database [15] compared to related work. For the voting stage our coverage score increases the performance by 6.5% [6], 8.5% [5] and 16.1% [13] leading to state-of-the-art voting results at reduced runtime.

Voting and Ranking Stage (FPPI=1.0)						Verification Stage (FPPI=0.3/0.4)					
ETHZ Classes	Hough [13]	$M^2HT$ [5]	$w_{ac}$ [6]	Our work	PMK [6]	Our work	$M^2HT$ [5]	PMK [6]	KAS [3]	System Full [13]	Our work
Apples	43.0	80.0	85.0	90.4	80.0	90.4	95.0/95.0	95.0/95.0	50.0/60.0	77.7/83.2	93.3/93.3
Bottles	64.4	92.4	67.0	84.4	89.3	96.4	92.9/96.4	89.3/89.3	92.9/92.9	79.8/81.6	97.0/97.0
Giraffes	52.2	36.2	55.0	50.0	80.9	78.8	89.6/89.6	70.5/75.4	49.0/51.1	39.9/44.5	79.2/81.9
Mugs	45.1	47.5	55.0	32.3	74.2	61.4	93.6/96.7	87.3/90.3	67.8/77.4	75.1/80.0	84.6/86.3
Swans	62.0	58.8	42.5	90.1	68.6	88.6	88.2/88.2	94.1/94.1	47.1/52.4	63.2/70.5	92.6/92.6
Average	53.3	63.0	60.9	<b>69.4</b>	78.6	<b>83.2</b>	91.9/93.2	87.2/88.8	61.4/66.9	67.2/72.0	<b>89.3/90.5</b>

horses (Section 4.2). We significantly outperform related methods in the hypotheses generation stage, while attaining competitive results for the full system. Results demonstrate that exploiting the connectedness of edge contours in a partial contour matching scenario enables to accurately localize category instances in images in efficient manner. Note also that we only use binary edge information for the hypothesis voting and do not include edge magnitude information, which plays important roles in other work [3,4,6,5].

Our proposed object localization method is not inherently scale invariant. We analyze 10 scales per image, where scale is defined by the distance between the sampled points. Localization of an object over all scales (!) requires on average only 5.3 seconds per image for ETHZ in a Matlab implementation.

#### 4.1 ETHZ Shape Classes

Results are reported on the challenging ETHZ shape dataset consisting of five object classes and a total of 255 images. All classes contain significant intra-class variations and scale changes. The images sometimes contain multiple instances of a category and have a large amount of background clutter.

Unfortunately direct comparison to related work is quite hard since many different test protocols exist. Foremost, on the ETHZ dataset there exist two main methods for evaluation. First, a class model is learned by training on half of the positive examples from a class, while testing is done on all remaining images (half of positive examples and all other negative classes) averaged over five random splits. Second, the ETHZ dataset additionally provides hand-drawn templates per class to model the categories. This step requires no training and has shown to provide slightly better results in a direct comparison [13]. Further, the detection performance may be evaluated using one of the two measures, namely the stricter PASCAL or the 20%-IoU criterion, which require that the intersection of the bounding box of the predicted hypotheses and the ground truth over the union of the two bounding boxes is larger than 50% or 20% respectively. Additional aspects in the evaluation are the use of 5-fold cross validation, aspect

ratio voting and most influential the use of features. Using strong features including color and appearance information naturally has a benefit over gradient information and again over pure binary shape information. This spectrum of features has the benefit to complement each other. Thus in our approach we use the hand-drawn models to match only binary edges in an query image and for a full system we further verify their location using a standard gradient-based classifier trained on half of the positive training samples.

Class-wise results for ETHZ using the strict PASCAL criterion are given in Table 1. The focus of this work lies on hypothesis voting stage, where we can show excellent results of 69.4% and 83.2%, without and with a PMK classifier ranking. The PMK ranking increases the scores for three classes (bottles, giraffes and mugs). The reason is that the classifier is better able to predict the instance of these classes, especially for mugs, where our system produces twice as many hypotheses compared to the other classes (on average 20 for mugs compared to 8 for the other classes). The coverage score performs better on compact object classes (applelogos and swans). Please note, the other methods do not use hand-drawn prototypes. We achieve an overall improvement over related work ranging from 6.5% [6], 8.5% [5] to 16.1% [13] without classifier ranking, and 4.6% over [6] using a classifier ranking. We also achieve competitive results after verification of 90.5% compared to 66.9% [3], 72.0% [13], 88.8% [6] and 93.2% [5] at 0.4 FPPI.

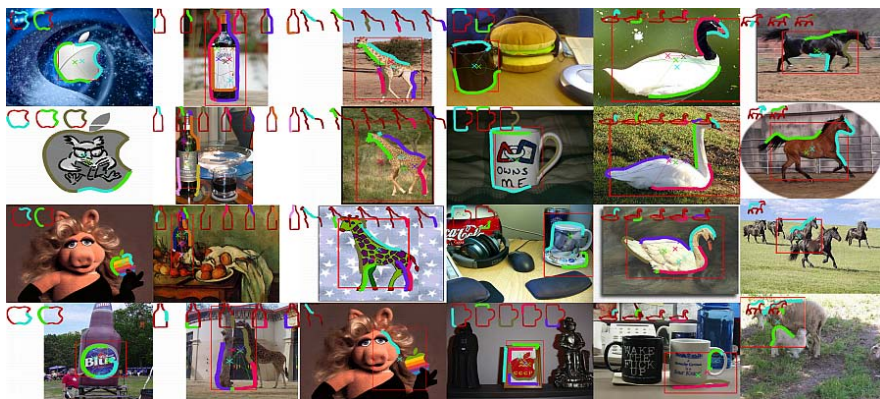
Due to the lack of hypothesis voting results for other approaches, we also provide a range of comparisons with previous work using the full system. We evaluate our method using the 20%-IoU criterion and summarize the results in Table 2. Compared to related work we also achieve excellent results using this criterion. Note again, that direct comparison has to be seen with caution, since methods either use hand-drawn or learned models. See Figure 5 for some exemplary successful detections and some failure cases.

**Table 2.** Average detection rates for related work on hand-drawn and learned models

ETHZ shape classes: <i>Verification Stage (FPPI = 0.3/0.4) using 20%-IoU</i>							
Method	Supervised	Template	Template	Template	Codebook	Learned	Template+Learned
	Lu [9]	Ravishankar [4]	Ferrari [15]	Ferrari [16]	Ferrari [3]	Ferrari [16]	<b>Our work</b>
Average	90.3/91.9	93.0/95.2	70.5/81.5	82.4/85.3	74.4/79.7	71.5/76.8	<b>94.4/95.2</b>

## 4.2 INRIA Horses

As a second dataset we use the INRIA horses [13], which consists of 170 images with one or more horses in side-view at several scales and cluttered background, and 170 images without horses. We use the same training and test split as [13] of 50 positive examples for the training and test on the remaining images (120+170). We again use only a single reference template which was chosen from the pixel-wise segmentation of a random horse from the training set. For this dataset the performance is 83.72% at FPPI=1.0 and thus is better than recent results 73.75% by [16], 80.77% by [3] and almost as good as 85.27% by [5], which additionally vote for aspect ratios. Presumably this would also increase our recall



**Fig. 5.** Results on ETHZ shape classes and INRIA horses (also see additional material)

for the strongly articulated horses since we detect the partial matches, however a single rigid reference template does not capture the centroid change.

## 5 Conclusion

We have presented a new approach in the paradigm of contour-based object detection based on partial contour matches to a reference template and show competitive results on state-of-the-art datasets like ETHZ shape and INRIA horses. Complementary to related work, we demonstrated that we can relax the approximations by piecewise segments by providing partial matching of contours instead of selecting or ignoring complete contours as well as extending the search beyond local neighborhoods of interest points. Our system implicitly handles parts of a contour and thus does not require grouping long salient curves or harmful splitting of contours to be able to match parts. Though a verification stage is a vital part for a full object detection system, we believe the focus should lie on better reflecting the hypotheses voting space, since this has a direct effect on the speed and accuracy of the full detector performance. In future work we will investigate learning discriminating weights [15] and interactions between contour fragments [17, 3].

## References

1. Shotton, J., Blake, A., Cipolla, R.: Contour-based learning for object detection. In: ICCV (2005)
2. 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. 3952, pp. 575–588. Springer, Heidelberg (2006)
3. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of adjacent contour segments for object detection. PAMI (2008)
4. Ravishankar, S., Jain, A., Mittal, A.: Multi-stage contour based detection of deformable objects. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 483–496. Springer, Heidelberg (2008)

5. Maji, S., Malik, J.: Object detection using a max-margin hough transform. In: CVPR (2009)
6. Ommer, B., Malik, J.: Multi-scale object detection by clustering lines. In: ICCV (2009)
7. Bai, X., Li, Q., Latecki, L., Liu, W., Tu, Z.: Shape band: A deformable object detection approach. In: CVPR (2009)
8. Zhu, Q., Wang, L., Wu, Y., Shi, J.: Contour context selection for object detection: A set-to-set contour matching approach. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 774–787. Springer, Heidelberg (2008)
9. Lu, C., Latecki, L., Adluru, N., Ling, H., Yang, X.: Shape guided contour fragment grouping with particle filters. In: ICCV (2009)
10. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: ICCV (2003)
11. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. PAMI (2005)
12. Biederman, I.: Human image understanding: Recent research and a theory. In: Computer Vision, Graphics, and Image Processing, vol. 32 (1985)
13. Ferrari, V., Jurie, F., Schmid, C.: From images to shape models for object detection. In: IJCV (2009)
14. Berg, A., Berg, T., Malik, J.: Shape matching and object recognition using low distortion correspondences. In: CVPR (2005)
15. 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. 3953, pp. 14–28. Springer, Heidelberg (2006)
16. Ferrari, V., Jurie, F., Schmid, C.: Accurate object detections with deformable shape models learnt from images. In: CVPR (2007)
17. Leordeanu, M., Hebert, M., Sukthankar, R.: Beyond local appearance: Category recognition from pairwise interactions of simple features. In: CVPR (2007)
18. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. PAMI (2002)
19. Gu, C., Lim, J., Arbelaez, P., Malik, J.: Recognition from regions. In: CVPR (2009)
20. Turney, J., Mudge, T., Volz, R.: Recognizing Partially Occluded Parts. PAMI (1985)
21. Brendel, W., Todorovic, S.: Video object segmentation by tracking regions. In: ICCV (2009)
22. Chen, L., Feris, R., Turk, M.: Efficient partial shape matching using smith-waterman algorithm. In: NORDIA (2008)
23. Felzenszwalb, P., Schwartz, J.: Hierarchical matching of deformable shapes. In: CVPR (2007)
24. Kokkinos, I., Yuille, A.: Hop: Hierarchical object parsing. In: CVPR (2009)
25. Donoser, M., Riemenschneider, H., Bischof, H.: Efficient partial shape matching of outer contours. In: Zha, H., Taniguchi, R.-i., Maybank, S. (eds.) ACCV 2009. LNCS, vol. 5994, pp. 281–292. Springer, Heidelberg (2009)
26. Donoser, M., Riemenschneider, H., Bischof, H.: Linked Edges as Stable Region Boundaries. In: CVPR (2010)
27. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. PAMI (2004)
28. Xu, C., Liu, J., Tang, X.: 2D Shape Matching by Contour Flexibility. PAMI (2009)
29. Grauman, K., Darrell, T.: The pyramid match kernel: Discriminative classification with sets of image features. In: ICCV (2005)

# Active Mask Hierarchies for Object Detection

Yuanhao Chen<sup>1</sup>, Long (Leo) Zhu<sup>2</sup>, and Alan Yuille<sup>1</sup>

<sup>1</sup> Department of Statistics, UCLA

<sup>2</sup> CSAIL, MIT

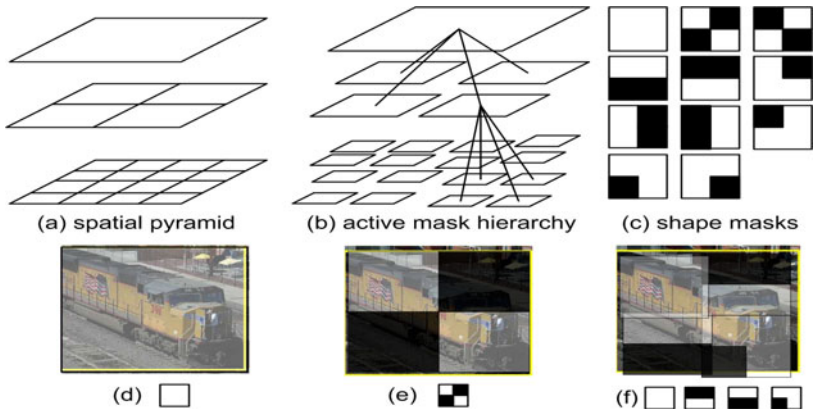
**Abstract.** This paper presents a new object representation, Active Mask Hierarchies (AMH), for object detection. In this representation, an object is described using a mixture of hierarchical trees where the nodes represent the object and its parts in pyramid form. To account for shape variations at a range of scales, a dictionary of masks with varied shape patterns are attached to the nodes at different layers. The shape masks are “active” in that they enable parts to move with different displacements. The masks in this active hierarchy are associated with histograms of words (HOWs) and oriented gradients (HOGs) to enable rich appearance representation of both structured (eg, cat face) and textured (eg, cat body) image regions. Learning the hierarchical model is a latent SVM problem which can be solved by the incremental concave-convex procedure (iCCCP). The resulting system is comparable with the state-of-the-art methods when evaluated on the challenging public PASCAL 2007 and 2009 datasets.

## 1 Introduction

The difficulty of object detection is because objects have complex appearance patterns and spatial deformations which can all occur at a range of different scales. Appearance patterns can be roughly classified into two classes: (i) structural (e.g., the head of a cat) which can be roughly described by the intensity edges and their spatial relations (e.g. by histogram of oriented gradients (HOGs)), and (ii) textural (e.g., the fur of a cat) which can be modeled by histograms of image features or words (e.g., histogram of words (HOWs)). Moreover these patterns can deform spatially both by translation – i.e., an entire image patch move – and/or by being partially masked out. The approach in this paper develops a novel Active Mask Hierarchy (AMH) which combines both types of appearance cues (HOWs and HOGs), allows subparts of the object to move actively and use a variety of different masks to deal with spatial deformations, and represents these appearance and geometric variations at a range of scales by a hierarchy.

Our work relates to two recent object representations which have made a significant impact in computer vision: (i) spatial pyramids [1], and (ii) part-based model [2]. But both approaches have strengths and weaknesses in the way that they deal with appearance variations and shape deformations. In this paper we seek an object representation which combines their strengths.





**Fig. 1.** (a) A spatial pyramid where the cells are bound together. (b) An Active Mask Hierarchy is represented by a tree structure where nodes are connected between consecutive layers and allowed to move object parts with displacements at different scales. (c) Shape masks include 11 generic shapes, such as vertical and horizontal bars, oriented L’s, etc. The white regions show the “valid” areas, where features are computed, while the black regions are “invalid”. The first mask is the rectangle used in standard spatial pyramids. (d) The train image example. A rectangle is used at the top layer to represent the entire object including some background. (e) Another (diagonal) mask is also used at the top layer to describe the train. (f) Four masks at the second level can be translated actively to better describe the object shape.

Spatial pyramids were proposed in [3] for scene classification and applied to object detection by [1]. A spatial pyramid is a three-layer pyramid, as shown in figure 1(a), where cells at different levels of the grid specify histograms of words (HOWs) located in the corresponding spatial domain yielding a coarse-to-fine representation. HOWs are particularly successful at modeling textured regions (e.g., a cat’s body), but are not well suited for describing structured regions (e.g., a cat’s face). Some papers [4, 1] use complementary descriptors, ie, histograms of oriented gradients (HOGs) [5], to account for other appearance variations. But two limitations still remain in the pyramid framework: (i) the cells are tightly bound spatially and are not allowed to move in order to deal with large spatial deformations of object parts (although pyramid of HOWs do tolerate a certain amount of spatial deformation). (ii) the cells have a rigid rectangular form and so are not well suited for dealing with partial overlaps of the object and its background. For example, the bounding box for the train in figure 1(d) includes cluttered background which makes HOWs less distinguishable.

Part-based models [2] are two-layer structures where the root node represent the entire object while the nodes at the second layers correspond to the parts. Unlike spatial pyramids, the nodes are allowed to move to account for large deformations of object parts. But part-based models also have two limitations. Firstly, the appearance models of the parts, which is based on HOGs [2], is not suitable for regions with rich texture properties where gradients are not

very informative. Secondly, the shallow structure (i.e., lack of a third layer) limits the representation of detailed appearance of the object and prevents the representation of small scale shape deformations.

This paper presents a new representation, called “Active Mask Hierarchies (AMH)”, which offers a richer way to represent appearance variations and shape deformations. Our approach combines spatial pyramids and part-based models into a single representation. First observe that an active mask hierarchy can be considered as a spatial pyramid with relaxed bonds – hence “active” (see fig. 1(b)). It can be represented by a tree structure where nodes at consecutive layers are vertically related, and assigned latent position variables to encode displacements of parts. Similarly active mask hierarchies can be thought of as a three-layer part-based model where “parts” together with their connections are simply designed as the active cells at different layers which are organized in a form of multi-level grids. As a result, the complicated procedure of part selection [2] is avoided. We will show that the multi-level grid design does not prevent us from achieving good performance.

Cells at different levels of the active mask hierarchy have appearance features based on HOGs and HOWs so as to model both structured and textured regions. Moreover, we assign a dictionary of masks with various binary shape patterns (fig. 1(c)) to all nodes which enable the part to deal with variations in the shape (i.e., overcome the restriction to regular rectangular templates). The features are only measured in the white areas specified by the masks. For example, masks (fig. 1(d) and 1(e)) at the top layer give the coarse descriptions of the boundary of the entire object. The active masks at the lower layers (fig. 1(f)) with displacements combine to represent the object parts more accurately. The selection of masks is performed by weighting their importance.

Learning the hierarchical model is a latent structural SVM problem [6] which can be solved by the concave-convex procedure (CCCP). CCCP has been successfully applied to learning models for object detection [7,8]. In order to reduce the training cost we use the variant called incremental concave-convex procedure (iCCCP) first reported in [8]. iCCCP allows us to learn hierarchical models using a large-scale training set efficiently.

Our experimental results demonstrate that the active mask hierarchies achieve state-of-the-art performance evaluated on the challenging public PASCAL 2007 and 2009 datasets [9,10]. As we show, the proposed method performs well at detecting both structured objects and textured objects.

## 2 Related Work

Hierarchical decomposition has also been explored in object recognition and image segmentation, such as [11,12,13]. Our use of shape masks is partially inspired by Levin and Weiss’s fragments [14], Torralba et al.’s spatial mask [15] and by Zhu et al.’s recursive segmentation templates [16]. But [14,16] are applied to segmentation and not to object detection. The masks used in [15] are not associated with latent position variable. The idea of “active” parts is similar in

spirit to Wu et al.’s active basis model [17], which does not involve a hierarchy. Schnitzspan et al.’s [18] uses a hierarchical models, but does not contain the shape masks.

There has been much related work on object detection, including [1,2,8,19,20]. [1,2,8] focus on the modeling of objects. Vedaldi et al. [1] present multiple kernel learning applied to spatial pyramids of histograms of features. They use a cascade of models and non-linear RBF kernels. Felzenszwalb et al. [2] propose latent SVM learning for part-based models and explore the benefit of post-processing (eg, incorporating contextual information). As we will show in the experiments, our system gives better performance without needing these “extras”. We use the iCCCP learning method developed in [8], but [8] does not use shape masks or HOWs (which give significant performance improvement as reported in the experimental section).

Instead of improving the representation of objects, both [19] and [20] focus on using global contextual cues to improve the performance of object detection. Desai et al. [19] make use a set of models from different object categories. [20] considers global image recognition and local object detection jointly.

### 3 Active Mask Hierarchies

In this section, we first formulate object learning as a latent structural SVM learning problem and then describe the active mask hierarchy representation. Finally, we briefly summarize the optimization method for training and the inference algorithm for detection.

#### 3.1 Active Mask Hierarchies and Latent Structural SVM

The goal of the AMH model is to detect whether an object with class label  $y$  is present in an image region  $x$ . The AMH model has latent variables  $h = (V, \mathbf{p})$  (i.e. not specified in the training set), where  $V$  labels the mixture component and  $\mathbf{p}$  specifies the positions of the object masks.

The AMH is specified by a function  $w \cdot \Phi(x, y, h)$  where  $w$  is a vector of parameter weights (to be learnt) and  $\Phi$  is a feature vector.  $\Phi$  has two types of terms: (i) appearance terms  $\Phi_A(x, y, h)$  which relate features of the image  $x$  to object classes  $y$ , components  $V$ , and mask positions  $\mathbf{p}$ ; (ii) shape terms  $\Phi_S(y, h)$  which specify the relationships between the positions of different masks and which are independent of the image  $x$ .

The *inference task* is to estimate the class label  $y$  and the latent states  $h$  by maximizing the discriminant function (assuming  $w$  is known):

$$F_w(x) = \operatorname{argmax}_{y, h} [w \cdot \Phi(x, y, h)] \quad (1)$$

The *learning task* is to estimate the optimal parameters  $w$  from a set of training data  $(x_1, y_1, h_1), \dots, (x_N, y_N, h_N)$ . We formulate the learning task as latent structural SVM learning. The object labels  $\{y_i\}$  of the image regions  $\{x_i\}$  are

known but the latent variables  $\{h_i\}$  are unknown (recall that the latent variables encode the mask positions  $\mathbf{p}$  and the model component  $V$ ). The task is to find the weights  $w$  which minimize an objective function  $J(w)$ :

$$J(w) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \left[ \max_{y,h} [w \cdot \Phi_{i,y,h} + L_{i,y,h}] - \max_h [w \cdot \Phi_{i,y_i,h}] \right] \quad (2)$$

where  $C$  is a fixed number,  $\Phi_{i,y,h} = \Phi(x_i, y, h)$  and  $L_{i,y,h} = L(y_i, y, h)$  is a loss function. For our object detection problem  $L(y_i, y, h) = 1$ , if  $y_i = y$ , and  $L(y_i, y, h) = 0$  if  $y_i \neq y$  (note  $L(\cdot)$  is independent of the latent variable  $h$ ).

Solving the optimization problem in equation (2) is difficult because the objective function  $J(w)$  is non-convex (because the fourth term  $-\max_h [w \cdot \Phi_{i,y_i,h}]$  is a concave function of  $w$ ). Following Yu and Joachims [6] we use the concave-convex procedure (CCCP) [21] which is guaranteed to converge at least to a local optimum. We note that CCCP has already been applied to learning models for object detection [7, 8]. We briefly describe CCCP and its application to latent SVMs in section 3.3.

In practice, the inner product in the discriminative function in equation (1) can be expressed as a summation of kernel functions [1]:

$$w \cdot \Phi(x, y, h) = \sum_{i,y',h'} \alpha_{i,y',h'} \mathcal{K}(\Phi_{i,y',h'}, \Phi_{x,y,h}) \quad (3)$$

where  $\alpha_{i,y',h'}$  are weights for support vectors obtained by solving equation (2) and  $\mathcal{K}(\Phi_{i,y',h'}, \Phi_{x,y,h})$  is a positive definite kernel, which can be represented by a linear (convex) combination of kernels:

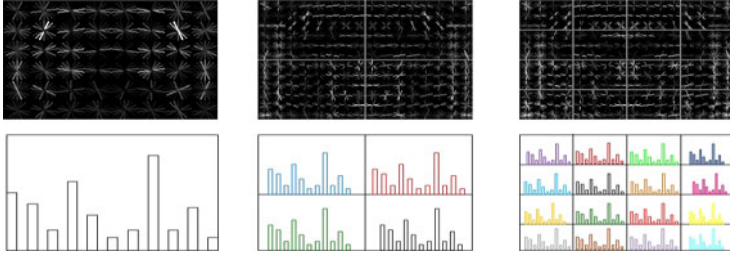
$$\mathcal{K}(\Phi_{i,y',h'}, \Phi_{x,y,h}) = \sum_k d_k \mathcal{K}_k(\Phi_{i,y',h'}, \Phi_{x,y,h}) \quad (4)$$

where  $\mathcal{K}_k(\Phi_{i,y',h'}, \Phi_{x,y,h})$  correspond to the appearance and shape kernels and  $d_k$  are their weights. We will introduce these kernels in section (3.2).

### 3.2 The Representation: Hierarchical Model and Feature Kernels

An AMH represents an object class by a mixture of two 3-layer tree-structured models. The structure of the model is shown in fig. 1(b). The structure used in our experiments is slightly different, but, for the sake of simplicity, we will use this structure to illustrate the basic idea and describe the difference in section 4.5.

The first layer has one root node which represents the entire object. The root node has four child nodes at the second layer in a  $2 \times 2$  grid layout where each cell represents one fourth of an object. Each node at the second layer has 4 child nodes at the third layer which contains 16 nodes in a  $4 \times 4$  grid layout. There are 21 ( $1 + 2 \times 2 + 4 \times 4$ ) nodes in total. Note that the cells in the spatial pyramid (figure 1(a)) are not connected.



**Fig. 2.** The top three panels show the Histogram of Oriented Gradients (HOGs). The bottom three panels show the Histogram Of Words (HOWs) extracted within different cells. The visual words are formed by using SIFT descriptors. The columns from left to right correspond to the top to bottom levels of the active hierarchy.

The numbers of layers and nodes are the same for different object classes and mixture components. But their aspect ratios may be different. Each tree model is associated with latent variables  $h = (V, \mathbf{p})$ .  $V \in \{1, 2\}$  is the index of the mixture components and  $\mathbf{p} = ((u_1, v_1), (u_2, v_2), \dots, (u_{21}, v_{21}))$  encodes the positions of all nodes. For an object class, let  $y = +1$  denote object and  $y = -1$  denote non-object. Let  $a \in \{1, \dots, 21\}$  index the nodes.  $b \in Ch(a)$  indexes the child nodes of node  $a$ .

The feature vector for each mixture component  $V$  is defined as follows:

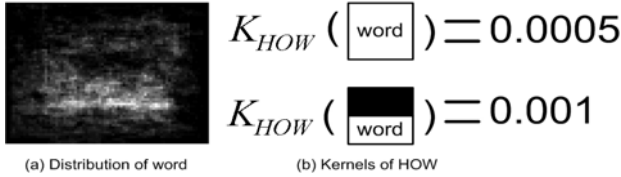
$$\Phi(x, y, \mathbf{p}) = \begin{cases} (\Phi_A(x, \mathbf{p}), \Phi_S(\mathbf{p})) & \text{if } y = +1 \\ 0 & \text{if } y = -1 \end{cases} \quad (5)$$

where  $\Phi_A(x, \mathbf{p})$  is a concatenation of appearance feature vectors  $\Phi_A(x, \mathbf{p}_a)$  which describe the image property of the corresponding regions specified by  $\mathbf{p}_a$ .  $\Phi_S(\mathbf{p})$  is a concatenation of shape feature vectors  $\Phi_S(\mathbf{p}_a, \mathbf{p}_b)$  which encode the parent-child spatial relationship of the nodes  $(\mathbf{p}_a, \mathbf{p}_b)$ . Note for different components  $V$ , we maintain separate feature vectors.

The appearance features consist of two types of descriptors (see fig. 2): (i) Histograms of Oriented Gradients (HOGs)  $\Phi_{HOG}(x, \mathbf{p})$  [5] and (ii) Histograms of Words (HOWs)  $\Phi_{HOW}(x, \mathbf{p})$  [4] extracted from SIFT descriptors [22] which are densely sampled. These two descriptors are complementary to each other for appearance representation. HOGs are suitable for structured regions where the image patches with specific oriented gradients (like car wheels, cat eyes, etc.) are located at certain position. On the other hand, HOW's advantages specialize at the textured regions where the small image patches encoded by visual words (like texon patches in cat body) appear randomly in a spatial domain. We followed the implementations of [2] to calculate the HOG descriptors, and [1] for the SIFT descriptors. Visual words are extracted by K-means using SIFT descriptors.

The HOW features are a vector of features calculated within the valid regions specified by the 11 shape masks, i.e.,

$$\Phi_{HOW}(x, \mathbf{p}) = \langle \Phi_{HOW}^1(x, \mathbf{p}), \Phi_{HOW}^2(x, \mathbf{p}), \dots, \Phi_{HOW}^{11}(x, \mathbf{p}) \rangle \quad (6)$$



**Fig. 3.** We illustrate the role of the mask and spatial variability (“active”) of the most important HOW feature at the top level of the AMH. Figure (a) plots the maximum response (for all masks) of visual word over the horse dataset. Observe that the response is peaked but has big spatial variability so that the AMH can adapt to spatial position and deformation of the objects. Figure (b), the most successful mask is the horizontal bar – mask 5, see figure (IIc) – which has, for example, twice as high kernel values as mask 1 (the regular rectangle).

The shape masks associated with node  $a$  are located by the latent position variable  $p_a$ . They are forms of varied binary shape patterns (fig. II(c)) which encode large shape variations of object and parts in a coarse-to-fine manner. The regions activated for the feature calculations are the white areas specified by the masks. For instance, the masks (fig. II(d) and II(e)) at the top layer give the coarse descriptions of the boundary of the entire object. The active masks at the lower layers (fig. II(f)) with displacements combine to represent the object parts more accurately. The patterns of shape masks are designed so that the histograms of words within the masks can be calculated efficiently using integral image.

$\Phi_S(h)$  is a concatenation of shape features  $\Phi_S(\mathbf{p}_a, \mathbf{p}_b), \forall a, b \in Ch(a)$ , which encode the parent-child pairwise spacial relationship. More precisely, the shape features for a parent-child pair  $(a, b)$  are defined as  $\Phi_S(\mathbf{p}_a, \mathbf{p}_b) = (\Delta u, \Delta v, \Delta u^2, \Delta v^2)$  where  $(\Delta u, \Delta v)$  is the displacement of node  $b$  relative to its reference position which is determined by the position of the parent node  $a$ . Our 3-layer model has 80 ( $4 \times 4 + 4 \times 16$ ) shape features in total.

Now we have complete descriptions of the appearance and shape features. The kernel in equation (4) which combines the appearance and shape kernels is given by (note we only consider the nontrivial case, i.e.,  $y = +1$ ):

$$\mathcal{K}(\Phi_{i,y',h'}, \Phi_{x,y,h}) = \mathcal{K}_A(\Phi(x_i, \mathbf{p}'), \Phi(x, \mathbf{p})) + \mathcal{K}_S(\Phi(\mathbf{p}'), \Phi(\mathbf{p})) \quad (7)$$

where  $\mathcal{K}_S(\Phi(\mathbf{p}'), \Phi(\mathbf{p}))$  is the shape kernel which is a simple linear kernel, i.e.  $\mathcal{K}_S(\cdot, \cdot) = \langle \Phi(\mathbf{p}'), \Phi(\mathbf{p}) \rangle$ .  $\mathcal{K}_A(\Phi(x_i, \mathbf{p}'), \Phi(x, \mathbf{p}))$  is the appearance kernel which is given by the weighted sum of two types of appearance kernels:

$$d_1 \mathcal{K}_1(\Phi_{HOG}(x_i, \mathbf{p}'), \Phi_{HOG}(x, \mathbf{p})) + d_2 \mathcal{K}_2(\Phi_{HOW}(x_i, \mathbf{p}'), \Phi_{HOW}(x, \mathbf{p})) \quad (8)$$

where  $d_1, d_2$  are weights for two appearance kernels respectively.  $\mathcal{K}_1(\cdot, \cdot)$  is a simple linear kernel, i.e.,  $\mathcal{K}_1(\cdot, \cdot) = \langle \Phi_{HOG}(x_i, \mathbf{p}'), \Phi_{HOG}(x, \mathbf{p}) \rangle$ .  $\mathcal{K}_2(\cdot, \cdot)$  is a quasi-linear kernel II, i.e.,  $\mathcal{K}_2(\cdot, \cdot) = \frac{1}{2}(1 - \mathcal{X}^2(\Phi_{HOW}(x_i, \mathbf{p}'), \Phi_{HOW}(x, \mathbf{p})))$ , which can be calculated efficiently using the technique proposed in [23]. Note that unlike II, the non-linear RBF kernels are not used here.

Figure (3) shows how the appearance kernels of the HOWs and the shape masks work. Recall that each HOW is computed for 11 masks, and the positions of these masks vary depending on the input image. Firstly, we explore the spatial variation of the maximum response of the HOW feature (for all masks) for the horse dataset. Our results, see figure (3a) show that the maximum response is spatially peaked in the lower center of the image window containing the object. But the position of the response varies considerably due to the variation in shape and location of the object. Secondly, by examining the mask kernel values, we see that mask 5 (horizontal bar) is the most effective when evaluated on this database and, see figure (3b), has kernel value which is twice as high as mask 1 (regular rectangle).

The free parameter in equation (8) is the ratio  $r$  of two weights  $d1 : d2$ . In our experiments, the ratio  $r$  is selected by cross validation as explored in (4). It is possible to improve the performance using more recent technique on feature combination (24). We leave it as future work.

Now we have a complete description for the representation of active mask hierarchies.

### 3.3 Optimization by CCCP

Learning the parameters  $w$  of the AMH model requires solving the optimization problem specified in equation (2). Following Yu and Joachims (6), we express the objective function  $J(w) = f(w) - g(w)$  where  $f(\cdot)$  and  $g(\cdot)$  are convex functions given by:

$$\begin{aligned} f(w) &= \left[ \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \max_{y,h} [w \cdot \Phi_{i,y,h} + L_{i,y,h}] \right] \\ g(w) &= \left[ C \sum_{i=1}^N \max_h [w \cdot \Phi_{i,y_i,h}] \right] \end{aligned} \quad (9)$$

The concave-convex procedure (CCCP) (21) is an iterative algorithm which converges to a local minimum of  $J(w) = f(w) - g(w)$ . When  $f(\cdot)$  and  $g(\cdot)$  take the forms specified by equation (9), then CCCP reduces to two steps (6) which estimate the latent variables and the model parameters in turn (analogous to the two steps of the EM algorithm):

Step (1): Estimate the latent variables  $h$  by the best estimates given the current values of the parameters  $w$ :  $h^* = (V^*, \mathbf{p}^*)$  (this is performed by the inference algorithm described in the following section).

Step (2): Apply structural SVM learning to estimate the parameters  $w$  using the current estimates of the latent variables  $h$ :

$$\min_w \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \left[ \max_{y,h} [w \cdot \Phi_{i,y,h} + L_{i,y,h}] - w \cdot \Phi_{i,y_i,h_i^*} \right] \quad (10)$$

We perform this structural SVM learning by the cutting plane method (25) to solve equation (10).

In this paper, we use a variant called *incremental CCCP* (iCCCP) first reported in [8]. The advantage of iCCCP is that it uses less training data and hence makes the learning more efficient. The kernel in equation (7) is applied without changing the training algorithm.

### 3.4 Detection: Dynamic Programming

The inference task is to estimate  $F_w(x) = \operatorname{argmax}_{y,h}[w \cdot \Phi(x, y, h)]$  as specified by equation (1). The parameters  $w$  and the input image region  $x$  are given. Inference is used both to detect objects after the parameters  $w$  have been learnt and also to estimate the latent variables during learning (Step 2 of CCCP).

The task is to estimate  $(y^*, h^*) = \operatorname{argmax}_{y,h}[w \cdot \Phi(x, y, h)]$ . The main challenge is to perform inference over the mask positions  $\mathbf{p}$  since the remaining variables  $y, V$  take only a small number of values. Our strategy is to estimate the  $\mathbf{p}$  by dynamic programming for all possible states of  $V$  and for  $y = +1$ , and then take the maximum. From now on we fix  $y, V$  and concentrate on  $\mathbf{p}$ .

First, we obtain a set of values of the root node  $\mathbf{p}_1 = (u_1, v_1)$  by exhaustive search over all subwindows at different scales of the pyramid. Next, for each location  $(u_1, v_1)$  of the root node we use dynamic programming to determine the best configuration  $\mathbf{p}$  of the remaining 20 parts. To do this we use the recursive procedure:

$$F(x, \mathbf{p}_a) = \sum_{b \in Ch(a)} \max_{\mathbf{p}_b} \{F(x, \mathbf{p}_b) + w \cdot \Phi_S(\mathbf{p}_a, \mathbf{p}_b)\} + w \cdot \Phi_A(x, \mathbf{p}_a) \quad (11)$$

where  $F(x, \mathbf{p}_a)$  is the max score of a subtree with root node  $a$ . The recursion terminates at the leaf nodes  $b$  where  $F(x, \mathbf{p}_b) = \Phi_A(x, \mathbf{p}_b)$ . This enables us to efficiently estimate the configurations  $\mathbf{p}$  which maximize the discriminant function  $F(x, \mathbf{p}_1) = \max_{\mathbf{p}} w \cdot \Phi(x, \mathbf{p})$  for each  $V$  and for  $y = +1$ .

The bounding box determined by the position  $(u_1, v_1)$  of the root node and the corresponding level of the image pyramid is output as an object detection if the score  $F(x, \mathbf{p}_1) >$  is greater than certain threshold.

In our implementations,  $w \cdot \Phi_A(x, \mathbf{p}_a)$  is replaced by the appearance kernel  $\mathcal{K}_A(\Phi(x_i, \mathbf{p}'), \Phi(x, \mathbf{p}))$  described in equation (8).

## 4 Experiments

The PASCAL VOC 2007 [9] and 2009 [10] datasets were used for evaluation and comparison. The PASCAL 2007 is the last version for which test annotations are available. There are 20 object classes which consist of 10000 images for training and testing. We follow the experimental protocols and evaluation criteria used in the PASCAL Visual Object Category detection contest 2007. A detection is considered correct if the intersection of its bounding box with the groundtruth bounding box is greater than 50% of their union. We compute precision-recall (PR) curves and score the average precision (AP) across a test set.



**Table 1.** Comparisons of performance on the PASCAL 2007 dataset. The numbers are the average precisions per category obtained by different methods. “UoCTTI-1” and “UoCTTI-2” report the results from [2] with and without special post-processing, respectively. “MKL-1” and “MKL-2” show the results obtained by [1] using quasi-linear kernels and RBF kernels, respectively.

Methods	Active Mask Hierarchies	no mask [8]	UoCTTI-1 [2]	UoCTTI-2 [2]	MKL-1 [1]	MKL-2 [1]	[19]	[20]	[18]
comments	HOG+HOW	HOG	Part-Based	+ context	pyramid	+RBF			
Ave. Precision	.338	.296	.268	.298	.291	.321	.271	.289	.275

**Table 2.** Performance Comparisons on the 20 PASCAL 2007 categories [9]. “Active Mask Hierarchies” refers to the proposed method in this paper. “UoCTTI-1” and “UoCTTI-2” report the results from [2] with and without special post-processing, respectively. “MKL-1” and “MKL-2” show the results obtained by [1] using quasi-linear kernels and RBF kernels, respectively. “V07” is the best result for each category among all methods submitted to the VOC 2007 challenge. Our method outperforms the other methods in 11 categories. The average APs per category of all methods are shown in the second column which have the corresponding numbers in table (1).

class	Ave.	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Active Mask Hierarchies (AMH)	<b>.338</b>	.348	.544	<b>.155</b>	.146	.244	<b>.509</b>	<b>.540</b>	<b>.335</b>	<b>.206</b>	.228
Hierarchy without masks [8]	.296	.294	.558	.094	.143	<b>.286</b>	.440	.513	.213	.200	.193
UoCTTI-1 (Part-based) [2]	.268	.290	.546	.006	.134	.262	.394	.464	.161	.163	.165
UoCTTI-2 (Part-based) [2]	.298	.328	<b>.568</b>	.025	<b>.168</b>	.285	.397	.516	.213	.179	.185
MKL-1 (Pyramid-based) [1]	.292	.366	.425	.128	.145	.151	.464	.459	.255	.144	.304
MKL-2 (Pyramid-based) [1]	.321	<b>.376</b>	.478	.153	.153	.219	.507	.506	.300	.173	<b>.330</b>
V07 [9]	—	.262	.409	.098	.094	.214	.393	.432	.240	.128	.140
class	Ave.	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Active Mask Hierarchies	<b>.338</b>	<b>.344</b>	<b>.241</b>	<b>.556</b>	<b>.473</b>	.349	<b>.181</b>	.202	<b>.303</b>	.413	.433
Hierarchy without masks [8]	.296	.252	.125	.504	.384	.366	.151	.197	.251	.368	.393
UoCTTI-1 (Part-based) [2]	.268	.245	.050	.436	.378	.350	.088	.173	.216	.340	.390
UoCTTI-2 (Part-based) [2]	.298	.259	.088	.492	.412	<b>.368</b>	.146	.162	.244	.392	.391
MKL-1 (Pyramid-based) [1]	.292	.190	.160	.490	.460	.215	.110	<b>.245</b>	.264	.426	.408
MKL-2 (Pyramid-based) [1]	.321	.225	.215	.512	.455	.233	.124	.239	.285	<b>.453</b>	<b>.485</b>
V07 [9]	—	.098	.162	.335	.375	.221	.120	.175	.147	.334	.289

#### 4.1 The Detection Results on the PASCAL Dataset

We compared our approach with other representative methods reported in the PASCAL VOC detection contest 2007 [9] and other more recent work [2, 11, 19, 20, 18]. Table (1) reports the Average Precisions per category (averaged over 20 categories) obtained by different methods. The comparisons in table (1) show that the active mask hierarchies (AMH) outperform other methods including state-of-the-art systems, i.e., [1] and [2].

It is important to realize that our result (0.338 AP) is obtained by a single model while all other methods’ final results rely on combining multiple models. For instance, “MKL-2” [1] (0.321 AP) uses cascade of models where non-linear RBF kernels and more features are used. “UoCTTI-2” [2] (0.298 AP) combines the detections output by models of all categories to access contextual information. It is clear that the additional processing improves the performance. For

**Table 3.** Performance Comparisons on the 20 PASCAL 2009 categories [10]. The approaches in the first column are described in table (2).

class	Ave.	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow
Active Mask Hierarchies (AMH)	.293	.432	.404	.135	.141	.271	.407	.355	.330	.172	.187
UoCTTI-2 (Part-based) [2]	.279	.395	.468	.135	.150	.285	.438	.372	.207	.149	.228
MKL-2 (Pyramid-based) [1]	.277	.478	.398	.174	.158	.219	.429	.277	.305	.146	.206
class	Ave.	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Active Mask Hierarchies	.293	.227	.219	.371	.444	.398	.129	.207	.247	.434	.342
UoCTTI-2 (Part-based) [2]	.279	.087	.144	.380	.420	.415	.126	.242	.158	.439	.335
MKL-2 (Pyramid-based) [1]	.277	.223	.170	.346	.437	.216	.102	.251	.166	.463	.376

example, the model with RBF kernels [1] improves by 0.03 AP and the post-processing used in [2] contributes 0.03 AP.

To give a better understanding how significant the improvement made by AMH is, three other recent advances are listed for comparisons. All of them explore the combination of multiple models as well. They achieve 0.271 ([19]), 0.289 ([20]) and 0.275([18]). [19] makes use of multiple models of different categories. [20] considers the recognition and detection jointly. [18] seeks to rescore the detection hypotheses output by [2].

In table (3), we report the performance evaluated on the PASCAL 2009 dataset. It also shows that our method is comparable with “MKL-2” [1] and “UoCTTI-2” [2]. In summary, our system built on a single model outperforms other alternative methods. It is reasonable to expect that our method with additional processing (e.g., RBF kernels, contextual cues, etc.) as used in other methods will achieve even better performance.

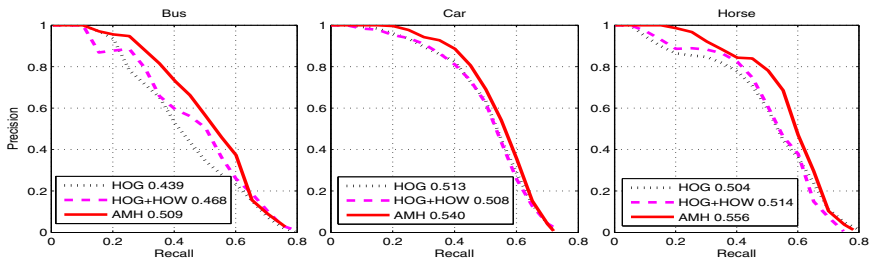
## 4.2 Active Mask Hierarchies, Spatial Pyramid and Part-Based Model

As we discussed before, spatial pyramid and part-based model can be unified in the representation of active mask hierarchies (AMH). It is of interest to study how differently (or similarly) each method performs in specific classes which have different scales of shape deformation and appearance variations. We show the detailed comparisons of the results on 20 object classes (PASCAL2007) in table (2). Our method obtains the best AP score in 11 out of 20 categories while MKL-2 using RBF kernels achieves the best performance in 4 categories. In order to show the advantage of the representation of AMH, it is more appropriate to compare AMH with MKL-1 which uses the same quasi-linear kernel of spatial pyramid, and “UoCTTI-1” which uses a part-based model only. Note AMH outperforms “UoCTTI-1” by 0.07 AP, “MKL-1” by 0.05 AP and [8] by 0.04 AP. Therefore, the improvement made by AMH is significant.

## 4.3 Benefit of Shape Masks

Table (1) shows that the active mask hierarchies (AMH) with both HOGs and HOWs outperform [8] by 0.04 AP. The detailed comparisons on 20 object classes

(PASCAL 2007) are shown in table (2). Recall that (8) uses HOGs only, and does not contain the shape masks and the HOW features. We quantify the gain contributed by HOWs and the shape masks. Figure (4) shows the PR curves of the three models using HOGs only (8), AMH (HOGs+HOWs) attached with one shape mask (regular rectangle), and AMH (HOGs+HOWs) with a dictionary of shape masks, respectively. HOWs improve the performance for bus by 0.03AP and horse by 0.01AP, but degrade the performance for car by less than 0.01AP. Adding shape masks makes improvement by 0.07, 0.03, 0.05 APs, for bus, car and horse, respectively.



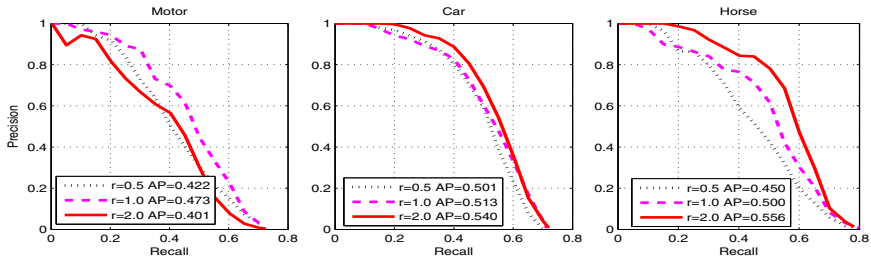
**Fig. 4.** The benefit of shape masks. “HOG” and “HOG+HOW” refer to the simple active hierarchy models without shape masks using HOGs only, and both HOGs and HOWs, respectively. “AMH” is the active shape hierarchy with both HOGs and HOWs. Three panels show the precision-recall curves evaluated on the bus, car, horse datasets.

#### 4.4 Weights of HOGs and HOWs

The ratio of weights of appearance kernels for HOGs and HOWs is selected by cross validation. Three values of the ratio  $r$ , i.e.,  $d1 : d2 = 0.5, 1.0, 2.0$  are tested. Figure (5) shows the PR curves of the models obtained by the appearance kernels with three values of  $r$ . For car, the result is less sensitive for the ratio, but for motorbike and horse, the maximum differences of performance are about 0.07 AP and 0.10 AP, respectively. The training cost is affordable if only one parameter needs to be selected. If more parameters are used, (24) can be used to learn the combination of appearance features in an efficient way.

#### 4.5 Implementation Details

All experiments are performed on a standard computer with a 3Ghz CPU.  $C$  is set to 0.005 for all classes. The detection time per image is 50 seconds. There are 300 visual words which are extracted by k-means where the color SIFT descriptors are used. The structure of the hierarchy used in our experiment is slightly different from the one shown in figure (1). In our implementations, the number of nodes at from top to bottom levels are  $1(1 \times 1), 9(3 \times 3), 36(6 \times 6)$ . The nodes are organized in regular multi-level grids. The HOW features  $\Phi_{HOW}$  at different layers of the pyramid are associated with fixed weights, i.e., 6:2:1,



**Fig. 5.** We compare the performance of AMHs with different ratios of weights of HOGs and HOWs. Three panels plot the precision-recall curves for the bus, car and horse datasets.

for all categories. As suggested by [4], other settings might further improve the performance. The settings of all free parameters used in the PASCAL 2007 and 2009 datasets are identical.

## 5 Conclusion

This paper describes a new active mask hierarchy model for object detection. This active hierarchy enables us to encode large shape deformation of object parts explicitly. The dictionary of masks with varied shape patterns increases our ability to represent shape and appearance variations. The active mask hierarchy uses histograms of words (HOWs) and oriented gradients (HOGs) to give rich appearance models for structured and textured image regions. The resulting system outperforms spatial pyramid and part-based models, and comparable with the state-of-the-art methods by evaluation on the PASCAL datasets.

**Acknowledgments.** Funding for this work was provided by NGA NEGI-1582-04-0004, MURI Grant N00014-06-1-0734, AFOSR FA9550-08-1-0489, NSF IIS-0917141 and gifts from Microsoft, Google and Adobe. Thanks to the anonymous reviewers for helpful feedback. We thank Antonio Torralba and Jianxiong Xiao for discussions.

## References

1. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: Proceedings of the International Conference on Computer Vision (2009)
2. Felzenszwalb, P.F., Grishick, R.B., McAllister, D., Ramanan, D.: Object detection with discriminatively trained part based models. PAMI (2009)
3. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2006)
4. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: CIVR (2007)

5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)
6. Yu, C.N.J., Joachims, T.: Learning structural svms with latent variables. In: International Conference on Machine Learning (ICML) (2009)
7. Vedaldi, A., Zisserman, A.: Structured output regression for detection with partial occlusion. In: Proceedings of Advances in Neural Information Processing Systems (NIPS) (2009)
8. Zhu, L., Chen, Y., Yuille, A., Freeman, W.: Latent hierarchical structural learning for object detection. In: CVPR (2010)
9. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge, VOC 2007 Results (2007), <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>
10. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge, VOC 2009 Results (2009), <http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html>
11. Epshtein, B., Ullman, S.: Feature hierarchies for object classification. In: Proceedings of IEEE International Conference on Computer Vision, pp. 220–227 (2005)
12. Zhu, S., Mumford, D.: A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision* 2, 259–362 (2006)
13. Storkey, A.J., Williams, C.K.I.: Image modelling with position-encoding dynamic. PAMI (2003)
14. Levin, A., Weiss, Y.: Learning to combine bottom-up and top-down segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 581–594. Springer, Heidelberg (2006)
15. Torralba, A., Murphy, K., Freeman, W.: Sharing visual features for multiclass and multiview object detection. PAMI (2007)
16. Zhu, L., Chen, Y., Lin, Y., Lin, C., Yuille, A.: Recursive segmentation and recognition templates for 2d parsing. In: Advances in Neural Information Processing Systems (2008)
17. Wu, Y.N., Si, Z., Fleming, C., Zhu, S.C.: Deformable template as active basis. In: ICCV (2007)
18. Schnitzspan, P., Fritz, M., Roth, S., Schiele, B.: Discriminative structure learning of hierarchical representations for object detection. In: Proc. CVPR (2009)
19. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for multi-class object layout. In: Proceedings of the International Conference on Computer Vision (2009)
20. Harzallah, H., Jurie, F., Schmid, C.: Combining efficient object localization and image classification. In: ICCV (2009)
21. Yuille, A.L., Rangarajan, A.: The concave-convex procedure (cccp). In: NIPS, pp. 1033–1040 (2001)
22. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60, 91–110 (2004)
23. Maji, S., Berg, A.C., Malik, J.: Classification using intersection kernel support vector machines is efficient. In: CVPR (2008)
24. Gehler, P., Nowozin, S.: On feature combination for multiclass object classification. In: ICCV (2009)
25. Tsochantaris, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: Proceedings of International Conference on Machine Learning (2004)

# From a Set of Shapes to Object Discovery

Nadia Payet and Sinisa Todorovic

Oregon State University,  
Kelley Engineering Center, Corvallis, OR 97331, USA  
payetn@onid.orst.edu, sinisa@eecs.oregonstate.edu

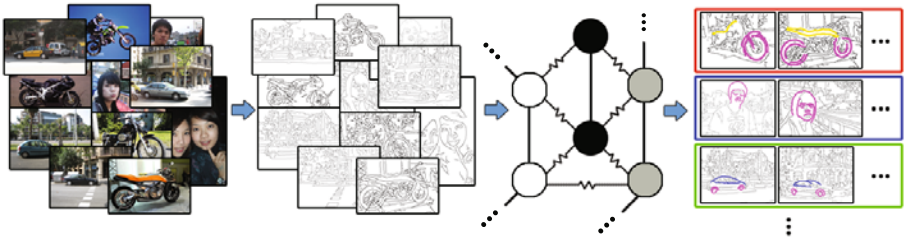
**Abstract.** This paper presents an approach to object discovery in a given unlabeled image set, based on mining repetitive spatial configurations of image contours. Contours that similarly deform from one image to another are viewed as collaborating, or, otherwise, conflicting. This is captured by a graph over all pairs of matching contours, whose maximum a posteriori multicoloring assignment is taken to represent the shapes of discovered objects. Multicoloring is conducted by our new Coordinate Ascent Swendsen-Wang cut (CASW). CASW uses the Metropolis-Hastings (MH) reversible jumps to probabilistically sample graph edges, and color nodes. CASW extends SW cut by introducing a regularization in the posterior of multicoloring assignments that prevents the MH jumps to arrive at trivial solutions. Also, CASW seeks to learn parameters of the posterior via maximizing a lower bound of the MH acceptance rate. This speeds up multicoloring iterations, and facilitates MH jumps from local minima. On benchmark datasets, we outperform all existing approaches to unsupervised object discovery.

## 1 Introduction

This paper explores a long-standing question in computer vision, that of the role of shape in representing and recognizing objects from certain categories occurring in images. In psychophysics, it is widely recognized that shape is one of the most categorical object properties [1]. Nevertheless, most recognition systems rather resort to appearance features (e.g., color, textured patches). Recent work combines shape with appearance features [2,3], but the relative significance of each feature type, and their optimal fusion for recognition still remains unclear.

Toward answering this fundamental question, we here focus on the problem of discovering and segmenting instances of frequently occurring object categories in arbitrary image sets. For object discovery, we use only the geometric properties of contour layouts in the images, deliberately disregarding appearance features. In this manner, our objective is to show that shape, on its own, without photometric features, is expressive and discriminative enough to provide robust detection and segmentation of common objects (e.g. faces, bikes, giraffes, etc.) in the midst of background clutter. To this end, we develop an approach to mining repetitive spatial configurations of contours across a given set of unlabeled images. As demonstrated in this paper, our shape mining indeed results in extracting (i.e., simultaneously detecting and segmenting) semantically meaningful objects recurring in the image set.

To our knowledge, this paper presents the first approach to extracting frequently occurring object contours from a clutter of image contours without any supervision,



**Fig. 1.** Overview: Given a set of unlabeled images (left), we extract their contours (middle left), and then build a graph of pairs of matching contours. Contour pairs that similarly deform from one image to another are viewed as collaborating (straight graph edges), or conflicting (zigzag graph edges), otherwise. Such coupling of contour pairs facilitates their clustering, conducted by our new algorithm, called Coordinate Ascent Swendsen-Wang cut (CASW). The resulting clusters represent shapes of discovered objects (right). (best viewed in color).

and without any help from appearance features. Existing work that uses only shape cues for recognition in real-world images requires either a manually specified shape template [4, 5], or manually segmented training images to learn the object shape [6]. Also, all previous work on unsupervised object-category discovery exploits the photometric properties of segments [7, 8], textured patches [9], and patches along image contours [10]. In our experiments, we outperform all these appearance-based, unsupervised approaches in both object detection and segmentation on benchmark datasets.

**Approach:** Our approach consists of three major steps, illustrated in Fig. 1. **Step 1:** Given a set of unlabeled images, we detect their contours by the minimum-cover algorithm of [11]. Each contour is characterized as a sequence of beam-angle descriptors, which are beam-angle histograms at points sampled along the contour. Similarity between two contours is estimated by the standard dynamic time warping (DTW) of the corresponding sequences of beam-angle descriptors. **Step 2** builds a weighted graph of matching contours, aimed at facilitating the separation of background from object shapes in Step 3. We expect that there will be many similarly shaped curves, belonging to the background in the images. Since the backgrounds vary, by definition, similar background curves will most likely have different spatial layouts across the image set. In contrast, object contours (e.g., curves delineating a giraffe’s neck) are more likely to preserve both shape and layout similarity in the set. Therefore, for object discovery, it is critical that we capture similar configurations of contours. To this end, in our graph, nodes correspond to pairs of matching contours, and graph edges capture spatial layouts of quadruples of contours. All graph edges can be both *positive* and *negative*, where their polarity is probabilistically sampled during clustering of image contours, performed in the next step. Positive edges support, and negative edges hinder the grouping of the corresponding contour pairs within the same cluster, if the contours *jointly* undergo similar (different) geometric transformation from one image to another. This provides stronger coupling of nodes than the common case of graph edges being only strongly or weakly “positive”, and thus leads to faster convergence to more accurate object discovery. **Step 3** conducts a probabilistic, iterative multicoloring of the graph,

by our new algorithm, called Coordinate-Ascent Swendsen-Wang (CASW) cut. In each iteration, CASW cut probabilistically samples graph edges, and then assigns colors to the resulting groups of connected nodes. The assignments are accepted by the standard Metropolis-Hastings (MH) mechanism. To enable MH jumps to better solutions with higher posterior distributions, we estimate parameters of the posterior by maximizing a lower bound of the MH acceptance rate. After convergence, the resulting clusters represent shapes of objects discovered, and simultaneously segmented, in the image set.

**Contributions:** Related to ours is the image matching approach of [12]. They build a similar graph of contours extracted from only two images, and then conduct multicoloring by the standard SW cut [13,12]. They pre-specify the polarity of graph edges, which remains fixed during multicoloring. Also, they hand-pick parameters of the posterior governing multicoloring assignments. In contrast, our graph is designed to accommodate transitive matches of many images, and we allow our graph edges to probabilistically change their polarity, in every MH iteration. We introduce a new regularization term in the posterior, which provides a better control of the probabilistic sampling of graph edges during MH jumps. Finally, we seek to *learn* parameters of our posterior via maximizing a lower bound of the MH acceptance rate. Our experiments show that this learning speeds up MH iterations, and allows jumps to solutions with higher posteriors.

Sec. 2 specifies our new shape descriptor. Sec. 3 describes how to build the graph from all pairs of image contours. Sec. 4 presents our new CASW cut for multicoloring of the graph. Sec. 5–6 present experimental evaluation, and our concluding remarks.

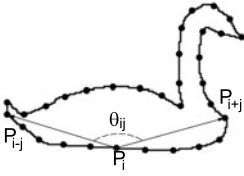
## 2 Image Representation Using Shapes and Shape Description

This section presents Step 1 of our approach. In each image, we extract relatively long, open contours using the minimum-cover algorithm of [11], referred to as gPb+ [11]. Similarity between two contours is estimated by aligning their sequences of points by the standard Dynamic Time Warping (DTW). Each contour point is characterized by our new descriptor, called weighted Beam Angle Histogram (BAH). BAH is a weighted version of the standard unweighted BAH, aimed at mitigating the uncertainty in contour extraction. BAH down-weights the interaction of distant shape parts, as they are more likely to belong to different objects in the scene.

The beam angles,  $\theta_{ij}$ , at contour points  $P_i$ ,  $i = 1, 2, \dots$ , are subtended by lines  $(P_{i-j}, P_i)$  and  $(P_i, P_{i+j})$ , as illustrated in Fig. 2.  $P_{i-j}$  and  $P_{i+j}$  are two neighboring points equally distant by  $j$  points along the contour from  $P_i$ ,  $j = 1, 2, \dots$ . BAH is a weighted histogram, where the weight of angle  $\theta_{ij}$  is computed as  $\exp(-\kappa j)$ ,  $j = 1, 2, \dots$  ( $\kappa = 0.01$ ). BAH is invariant to translation, in-plane rotation, and scale. Experimentally, we find that BAH with 12 bins gives optimal and stable results.

Table 1 compares BAH with other popular shape descriptors on the task of contour matching. We match contours from all pairs of images belonging to the same class in the ETHZ dataset [3], and select the top 5% best matches. True positives (false positives) are pixels of the matched contour that fall in (outside of) the bounding box of the target object. The ground truth is determined from pixels of the initial set of detected contours that fall inside the bounding box. For matching, we use DTW, and Oriented Chamfer Distance [2]. Tab. 1 shows that our BAH descriptor gives the best performance with all





**Fig. 2.** BAH is a weighted histogram of beam angles  $\theta_{ij}$  at contour points  $P_i$ ,  $i=1, 2, \dots$

**Table 1.** Contour matching on the ETHZ image dataset [3]. Top is *Precision*, bottom is *Recall*. The rightmost column shows matching results of Oriented Chamfer Distance [2], and other columns show DTW results. Descriptors (left to right): our BAH, unweighted BAH, Shape Context [14], and SIFT [15].

Contour detectors	BAH	BAH-U	[14]	[15]	[2]
Canny	$0.23 \pm 0.01$	0.21	0.18	0.15	0.21
	$0.59 \pm 0.02$	0.57	0.48	0.48	0.52
[3]	$0.32 \pm 0.03$	0.30	0.25	0.18	0.29
	$0.78 \pm 0.03$	0.75	0.62	0.61	0.72
gPb+ [11]	$0.37 \pm 0.02$	0.34	0.26	0.20	0.34
	$0.81 \pm 0.03$	0.78	0.63	0.61	0.74

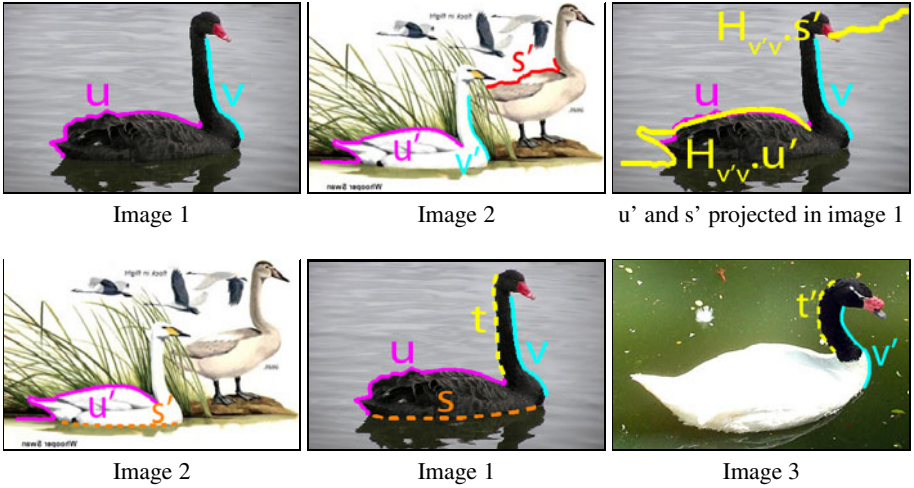
contour detectors, and the highest accuracy with gPb+ [11]. Also, DTW with our BAH outperforms Oriented Chamfer Distance.

### 3 Constructing the Graph of Pairs of Image Contours

This section presents Step 2 which constructs a weighted graph,  $G = (V, E, \rho)$ , from contours extracted from the image set. Nodes of  $G$  represent candidate matches of contours,  $(u, u') \in V$ , where  $u$  and  $u'$  belong to different images. Similarity of two contours is estimated by DTW. We keep only the best 5% of contour matches as nodes of  $G$ .

Edges of  $G$ ,  $e = ((u, u'), (v, v')) \in E$ , capture spatial relations of corresponding image contours. If contours  $u$  and  $v$  in image 1, and their matches  $u'$  and  $v'$  in image 2 have similar spatial layout, then they are less likely to belong to the background clutter. All such contour pairs will have a high probability to become positively coupled in  $G$ . Otherwise, matches  $(u, u')$  and  $(v, v')$  will have a high probability to become negatively coupled in  $G$ , so that CASW could place them in different clusters. This probabilistic coupling of nodes in  $G$  is encoded by edge weights,  $\rho_e$ , defined as the likelihood  $\rho_e^+ \propto \exp(-w_\delta^+ \delta_e)$ , given the positive polarity of  $e$ , and  $\rho_e^- \propto \exp(-w_\delta^- (1 - \delta_e))$ , given the negative polarity of  $e$ .  $w_\delta^+$  and  $w_\delta^-$  are the parameters of the exponential distribution, and  $\delta_e \in [0, 1]$  measures a difference in spatial layouts of  $u$  and  $v$  in image 1, and their matches  $u'$  and  $v'$  in image 2. We specify  $\delta_e$  for the following two cases. In Cases 1 and 2, there are at least two contours that lie in the same image. This allows establishing geometric transforms between  $((u, u'), (v, v'))$ . Note that this would be impossible, in a more general case, where  $((u, u'), (v, v'))$  come from four distinct images.

**Case 1:**  $(u, u')$  and  $(v, v')$  come from *two* images, where  $u$  and  $v$  are in image 1, and  $u'$  and  $v'$  are in image 2, as illustrated in Fig. 3a. We estimate  $\delta_e$  in terms of affine homographies between the matching contours, denoted as  $H_{uu'}$ , and  $H_{vv'}$ . Note that if  $u, v$  in image 1 preserve that same spatial layout in image 2, then  $H_{vv'} = H_{vu} H_{uu'}$ . Since the estimation of  $H_{vu}$  between arbitrary, non-similar contours  $u$  and  $v$  in image 1 is difficult, we use the following strategy. From the DTW alignment of points along  $u$  and  $u'$ , we estimate their affine homography  $H_{uu'}$ . Similarly, for  $v$  and  $v'$ , we estimate  $H_{vv'}$ . Then, we project  $u'$  to image 1, as  $u'' = H_{vv'} u'$ , and, similarly, project  $v'$  to image 1 as  $v'' = H_{uu'} v'$  (Fig. 3a right). Next, in image 1, we measure distances between



**Fig. 3.** (a) Case 1: Estimating  $\delta_{(u,u',v,v')}$  when contours  $u$  and  $v$  are in image 1, and their matches  $u'$  and  $v'$  are in image 2. We use the affine-homography projection of  $u'$  and  $v'$  to image 1,  $u'' = H_{vv'}u'$  and  $v'' = H_{uu'}v'$ , and compute  $\delta$  as the average distance between  $u$  and  $u''$ , and  $v$  and  $v''$ . As can be seen, pairs  $(u, s')$  and  $(v, v')$  do not have similar layouts in image 1 and image 2. (b) Case 2: Estimating  $\delta_{(u,u',v,v')}$  when  $u$  and  $v$  are in image 1, and their matches  $u'$  and  $v'$  are in image 2 and image 3. We use multiple affine-homography projections of  $u'$  and  $v'$  to image 1 via auxiliary, context contours  $s'$  and  $t'$  in a vicinity of  $u'$  and  $v'$ .

corresponding points of  $u$  and  $u''$ , where the point correspondence is obtained from DTW of  $u$  and  $u'$ . Similarly, we measure distances between corresponding points of  $v$  and  $v''$ .  $\delta_e$  is defined as the average point distance between  $u$  and  $u''$ , and  $v$  and  $v''$ .

**Case 2:**  $(u, u')$  and  $(v, v')$  come from *three* images, where  $u$  and  $v$  belong to image 1,  $u'$  is in image 2, and  $v'$  is in image 3. In this case, we can neither use  $H_{vv'}$  to project  $u'$  from image 2 to image 1, nor  $H_{uu'}$  to project  $v'$  from image 3 to image 1. Instead, we resort to context provided by auxiliary contours  $s'$  in a vicinity of  $u'$ , and auxiliary contours  $t'$  in a vicinity of  $v'$ . For every neighbor  $s'$  of  $u'$  in image 2, we find its best DTW match  $s$  in image 1, and compute homography  $H_{ss'}$ . Similarly, for every neighbor  $t'$  of  $v'$  in image 3, we find its best DTW match  $t$  in image 1, and compute homography  $H_{tt'}$ . Then, we use all these homographies to project  $u'$  to image 1, multiple times, as  $u''_s = H_{ss'}u'$ , and, similarly, project  $v'$  to image 1, multiple times, as  $v''_t = H_{tt'}v'$ . Next, as in Case 1, we measure distances between corresponding points of all  $u$  and  $u''_s$  pairs, and all  $v$  and  $v''_t$  pairs.  $\delta_e$  is defined as the average point distance.

## 4 Coordinate-Ascent Swendsen-Wang Cut

This section presents Step 3. Given the graph  $G = (V, E, \rho)$ , specified in the previous section, our goal is to perform multicoloring of  $G$ , which will partition  $G$  into two subgraphs. One subgraph will represent a composite cluster of nodes, consisting of a number of connected components (CCPs), receiving distinct colors, as illustrated in

Fig. 4. This composite cluster contains contours of the discovered object categories. Nodes outside of the composite cluster are interpreted as the background. All edges  $e \in E$  can be negative and positive. A negative edge indicates that the nodes are conflicting, and thus should not be assigned the same color. A positive edge indicates that the nodes are collaborative, and thus should be favored to get the same color. If nodes are connected by positive edges, they form a CCP, and receive the same color (Fig. 4). A CCP cannot contain a negative edge. CCPs connected by negative edges form a composite cluster. The amount of conflict and collaboration between two nodes is defined by the likelihood  $\rho$ , defined in Sec. 3.

For multicoloring of  $G$ , we formulate a new Coordinate Ascent Swendsen-Wang cut (CASW) that uses the iterative Metropolis-Hastings algorithm. CASW iterates the following three steps: (1) Sample a composite cluster from  $G$ , by probabilistically cutting and sampling positive and negative edges between nodes of  $G$ . This results in splitting and merging nodes into a new configuration of CCPs. (2) Assign new colors to the resulting CCPs within the selected composite cluster, and use the Metropolis-Hastings (MH) algorithm to estimate whether to accept this new multicoloring assignment of  $G$ , or to keep the previous state. (3) If the new state is accepted, go to step (1); otherwise, if the algorithm converged, re-estimate parameters of the pdf's controlling the MH iterations, and go to step (1), until the pdf re-estimation does not affect convergence.

CASW is characterized by large MH moves, involving many strongly-coupled graph nodes. This typically helps avoid local minima, and allows fast convergence, unlike other related MCMC methods. In comparison with [12], our three key contributions include: (a) the on-line learning of parameters of pdf's governing MH jumps; (b) enforcing stronger node coupling by allowing the polarity of edges to be dynamically estimated during the MH iterations; and (c) regularizing the posterior of multicoloring assignments to help MH jumps escape from trivial solutions. In the following, we present our Bayesian formulation of CASW, inference, and learning.

**Bayesian Formulation:** Multi-coloring of  $G$  amounts to associating labels  $l_i$  to nodes in  $V$ ,  $i=1, \dots, |V|$ , where  $l_i \in \{0, 1, \dots, K\}$ .  $K$  denotes the total number of target objects, which is a priori unknown, and  $(K + 1)$ th label is the background. The multicoloring solution can be formalized as  $\mathcal{M}=(K, \{l_i\}_{i=1, \dots, |V|})$ . To find  $\mathcal{M}$ , we maximize the posterior distribution  $p(\mathcal{M}|G)$ , as

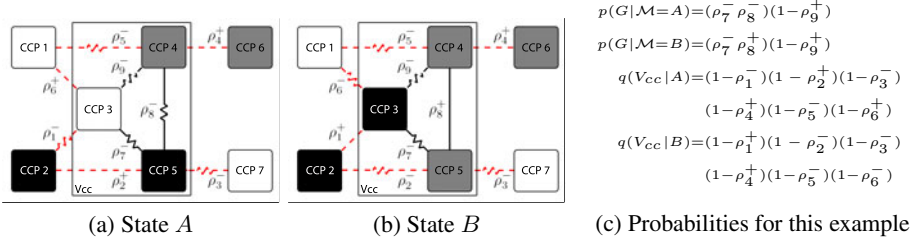
$$\mathcal{M}^* = \arg \max_{\mathcal{M}} p(\mathcal{M}|G) = \arg \max_{\mathcal{M}} p(\mathcal{M})p(G|\mathcal{M}). \quad (1)$$

Let  $N$  denote the number of nodes that are labeled as background  $l_i = 0$ . Also, let binary functions  $\mathbb{1}_{l_i \neq l_j}$  and  $\mathbb{1}_{l_i = l_j}$  indicate whether node labels  $l_i$  and  $l_j$  are different, and the same. Then, we define the prior  $p(\mathcal{M})$  and likelihood  $p(G|\mathcal{M})$  as

$$p(\mathcal{M}) \propto e^{-w_K K} e^{-w_N N}, \quad (2)$$

$$p(G|\mathcal{M}) \propto \prod_{e \in \mathbb{E}^+} \rho_e^+ \prod_{e \in \mathbb{E}^-} \rho_e^- \prod_{e \in \mathbb{E}^0} (1 - \rho_e^+) \mathbb{1}_{l_i \neq l_j} \cdot (1 - \rho_e^-) \mathbb{1}_{l_i = l_j}, \quad (3)$$

where  $p(\mathcal{M})$  penalizes large  $K$  and  $N$ .  $w_K$  and  $w_N$  are the parameters of the exponential distribution.  $\mathbb{E}^+$  and  $\mathbb{E}^-$  denote positive and negative edges present in the composite cluster, and  $\mathbb{E}^0$  denotes edges that are probabilistically cut (i.e., not present in the solution). Our  $p(G|\mathcal{M})$ , defined in (3), differs from the likelihood defined in [12]. In [12],



**Fig. 4.** (a) In state  $A$ , probabilistically sampled positive (straight bold) and negative (zigzag bold) edges define composite cluster  $V_{cc} = \{CCP3, CCP4, CCP5\}$  (cut edges are dashed). The cut is a set of edges (red) that have not been probabilistically sampled, which would otherwise connect  $V_{cc}$  to external CCPs. (b) The coloring of CCPs within  $V_{cc}$  is randomly changed, resulting in new state  $B$ . This also changes the type of edges  $\rho_1, \rho_2, \rho_6, \rho_8$ , since the positive (negative) edge may link only two CCPs with the same (different) label(s). (c) Probabilities in states  $A$  and  $B$ .

nodes can be connected by only one type of edges. They pre-select a threshold on edge weights, which splits the edges into positive and negative, and thus define the likelihood as  $p(G|\mathcal{M}) \propto \prod_{e \in \mathbb{E}^+} \rho_e^+ \prod_{e \in \mathbb{E}^-} \rho_e^-$ . Since we allow both types of edges to connect every pair of nodes, where the right edge type gets probabilistically sampled in every MH iteration, we enforce a stronger coupling of nodes. As shown in Sec. 5 this advanced feature of our approach yields faster convergence and better clustering performance. This is because our formulation maximizes the likelihood  $p(G|\mathcal{M})$  when every two nodes with the same label are (i) connected by a strong positive edge ( $e \in E^+$ , and  $\rho_e^+$  large), or (ii) remain unconnected, but the likelihood that these nodes should not have the same label is very low ( $e \in E^0$ , and  $\rho_e^-$  small). Similarly, our likelihood  $p(G|\mathcal{M})$  is maximized when every two nodes with different labels are (i) connected by a strong negative edge ( $e \in \mathbb{E}^-$ , and  $\rho_e^-$  large), or (ii) remain unconnected, but the likelihood that these nodes should have the same label is very low ( $e \in E^0$ , and  $\rho_e^+$  small).

**Inference:** We here explain the aforementioned iterative steps (1) and (2) of our CASW cut. Fig. 4 shows an illustrative example. In step (1), edges of  $G$  are probabilistically sampled. If two nodes have the same label, their positive edge is sampled, with likelihood  $\rho_e^+$ . Otherwise, if the nodes have different labels, their negative edge is sampled, with likelihood  $\rho_e^-$ . This re-connects all nodes into new connected components (CCPs). The negative edges that are sampled will connect CCPs into a number of composite clusters, denoted by  $V_{cc}$ . This configuration is referred to state  $A$ . In step (2), we choose at random one composite cluster,  $V_{cc}$ , and probabilistically reassign new colors to the CCPs within  $V_{cc}$ , resulting in a new state  $B$ . Note that all nodes within one CCP receive the same label, which allows large moves in the search space.

The CASW accepts the new state  $B$  as follows. Let  $q(A \rightarrow B)$  be the proposal probability for moving from state  $A$  to  $B$ , and let  $q(B \rightarrow A)$  denote the reverse. The acceptance rate,  $\alpha(A \rightarrow B)$ , of the move from  $A$  to  $B$  is defined as

$$\alpha(A \rightarrow B) = \min \left( 1, \frac{q(B \rightarrow A)p(\mathcal{M} = \mathcal{B}|\mathcal{G})}{q(A \rightarrow B)p(\mathcal{M} = \mathcal{A}|\mathcal{G})} \right). \quad (4)$$

Note that complexity of each move is relatively low, since computing  $\frac{q(B \rightarrow A)}{q(A \rightarrow B)}$  involves only those edges that are probabilistically cut around  $V_{cc}$  in states  $A$  and  $B$  — not all edges. Also,  $\frac{p(\mathcal{M}=\mathcal{B}|\mathcal{G})}{p(\mathcal{M}=\mathcal{A}|\mathcal{G})}$  accounts only for the recolored CCPs in  $V_{cc}$  — not the entire graph  $G$ . Below, we derive  $\frac{q(B \rightarrow A)}{q(A \rightarrow B)}$  and  $\frac{p(\mathcal{M}=\mathcal{B}|\mathcal{G})}{p(\mathcal{M}=\mathcal{A}|\mathcal{G})}$ , and present a toy example (Fig. 4).

$q(A \rightarrow B)$  is defined as a product of two probabilities: (i) the probability of generating  $V_{cc}$  in state  $A$ ,  $q(V_{cc}|A)$ ; and (ii) the probability of recoloring the CCPs within  $V_{cc}$  in state  $B$ , where  $V_{cc}$  is obtained in state  $A$ ,  $q(B(V_{cc})|V_{cc}, A)$ . Thus, we have  $\frac{q(B \rightarrow A)}{q(A \rightarrow B)} = \frac{q(V_{cc}|B)q(A(V_{cc})|V_{cc}, B)}{q(V_{cc}|A)q(B(V_{cc})|V_{cc}, A)}$ . The ratio  $\frac{q(A(V_{cc})|V_{cc}, B)}{q(B(V_{cc})|V_{cc}, A)}$  can be canceled out, because the CCPs within  $V_{cc}$  are assigned colors under the uniform distribution. Let  $\text{Cut}_A^+$  and  $\text{Cut}_A^-$  ( $\text{Cut}_B^+$  and  $\text{Cut}_B^-$ ) denote positive and negative edges which are probabilistically “cut” around  $V_{cc}$  in state  $A$  (state  $B$ ). Since the probabilities of cutting the positive and negative edges are  $(1-\rho_e^+)$  and  $(1-\rho_e^-)$ , we have

$$\frac{q(B \rightarrow A)}{q(A \rightarrow B)} = \frac{q(V_{cc}|B)}{q(V_{cc}|A)} = \frac{\prod_{e \in \text{Cut}_B^+} (1-\rho_e^+) \prod_{e \in \text{Cut}_B^-} (1-\rho_e^-)}{\prod_{e \in \text{Cut}_A^+} (1-\rho_e^+) \prod_{e \in \text{Cut}_A^-} (1-\rho_e^-)}. \quad (5)$$

For the example shown in Figure 4, we compute  $\frac{q(B \rightarrow A)}{q(A \rightarrow B)} = \frac{(1-\rho_1^+)(1-\rho_2^-)(1-\rho_6^-)}{(1-\rho_1^-)(1-\rho_2^+)(1-\rho_6^+)}$ .

Also,  $\frac{p(\mathcal{M}=\mathcal{B}|\mathcal{G})}{p(\mathcal{M}=\mathcal{A}|\mathcal{G})} = \frac{p(\mathcal{M}=\mathcal{B})p(G|\mathcal{M}=\mathcal{B})}{p(\mathcal{M}=\mathcal{A})p(G|\mathcal{M}=\mathcal{A})}$  can be efficiently computed.  $p(\mathcal{M}=\mathcal{B})$  can be directly computed from the new coloring in state  $B$ , and  $\frac{p(G|\mathcal{M}=\mathcal{B})}{p(G|\mathcal{M}=\mathcal{A})}$  depends only on those edges that have changed their polarity. For the example shown in Fig 4, we compute  $\frac{p(\mathcal{M}=\mathcal{B}|\mathcal{G})}{p(\mathcal{M}=\mathcal{A}|\mathcal{G})} = \frac{\rho_8^+}{\rho_8^-}$ .

When  $\alpha(A \rightarrow B)$  has a low value, and new state  $B$  cannot be accepted by MH, CD-SW remains in state  $A$ . In the next iteration, CD-SW either probabilistically selects a different  $V_{cc}$ , or proposes a different coloring scheme for the same  $V_{cc}$ .

**Learning:** Our Bayesian model is characterized by a number of parameters that we seek to learn from data. We specify that learning occurs at a standstill moment when MH stops accepting new states (we wait for 100 iterations). In that moment, the previous state  $A$  is likely to have the largest pdf in this part of the search space. By learning new model parameters, our goal is to allow for larger MH moves, and thus facilitate exploring other parts of the search space characterized by higher posterior distributions  $p(\mathcal{M}|G)$ . Since the moves are controlled by  $\alpha(A \rightarrow B)$ , given by (4), we learn the parameters by maximizing a lower bound of  $\alpha(A \rightarrow B)$ . If this learning still does not result in accepting new states, we conclude that the algorithm has converged.

From (3) and (4), and the definitions of edge likelihoods  $\rho_e^+$  and  $\rho_e^-$  given in Sec. 3, we derive a lower bound of  $\log(\alpha(A \rightarrow B))$  as

$$\log(\alpha(A \rightarrow B)) \geq \phi^T \mathbf{w}, \quad (6)$$

where  $\mathbf{w} = [w_K, w_N, w_\delta^+, w_\delta^-]^T$ , and  $\phi = [\phi_1, \phi_2, \phi_3, \phi_4]^T$  is the vector of observed features, defined as  $\phi_1 = K_A - K_B$ ,  $\phi_2 = N_A - N_B$ ,  $\phi_3 = \sum_{e \in \mathbb{E}_A^+} \delta_e - \sum_{e \in \mathbb{E}_B^+} \delta_e$ , and  $\phi_4 = \sum_{e \in \mathbb{E}_A^-} (1-\delta_e) - \sum_{e \in \mathbb{E}_B^-} (1-\delta_e)$ .  $\mathbb{E}_B^+$  denotes all edges in state  $B$  whose likelihood is  $\rho_+$ ,  $\mathbb{E}_B^+ = \mathbb{E}_B^+ \cup \text{Cut}_B^+ \cup \mathbb{E}_B^0$ , and  $\mathbb{E}_B^-$  denotes all edges in state  $B$  whose

likelihood is  $\rho^-$ ,  $\tilde{\mathbb{E}}_B^- = \mathbb{E}_B^- \cup \text{Cut}_B^- \cup \mathbb{E}_B^{0+}$ . From (6), we formulate learning as the following linear program

$$\max_{\mathbf{w}} \phi^T \mathbf{w}, \quad \text{s.t.} \quad \|\mathbf{w}\|_2 = 1, \quad (7)$$

which has a closed-form solution [16],  $\mathbf{w} = \frac{1}{\|\phi_+\|} \phi_+$ , where  $(\phi)_+ = \max(0, \phi)$ .

## 5 Results

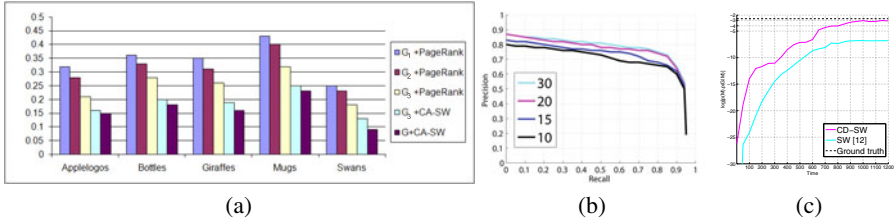
Given a set of images, we perform object discovery in two stages, as in [9, 17, 10]. We first coarsely cluster images based on their contours using CASW cut, and then again use CASW to cluster contours from only those images that belong to the same coarse cluster. The first stage serves to discover different object categories in the image set, whereas the second, fine-resolution stage serves to separate object contours from background clutter, and also extract characteristic parts of each discovered object category.

We use the following benchmark datasets: Caltech-101 [18], ETHZ [3], LabelMe [19], and Weizmann Horses [20]. In the experiments on Caltech-101, we use all Caltech images showing the same categories as those used in [9]. Evaluation on ETHZ and Weizmann Horses uses the entire datasets. For LabelMe, we keep the 15 first images retrieved by keywords *car side*, *car rear*, *face*, *airplane* and *motorbike*. ETHZ and LabelMe increase complexity over Caltech-101, since their images contain multiple object instances, which may: (a) appear at different resolutions, (b) have low contrasts with textured background, and (c) be partially occluded. The Weizmann Horses are suitable to evaluate performance on articulated, non-rigid objects.

We study two settings S1 and S2. In S1, we use only ETHZ to generate the input image set. The set consists of positive and negative examples, where positive images show a unique category, and negative ones show objects from other categories in ETHZ. In S2, the image set contains examples of all object categories from the considered dataset. S1 is used for evaluating particular contributions of our approach, and S2 is used for evaluating our overall performance.

In the first stage of object discovery, CASW finds clusters of images. This is evaluated by *purity*. Purity measures the extent to which a cluster contains images of a single dominant object category. When running CASW in the second stage, on each of these image clusters, we use *Bounding Box Hit Rate* (BBHR) to verify whether contours detected by CASW fall within the true foreground regions. The ground truth is defined as all pixels of the extracted image contours that fall in the bounding boxes or segments of target objects. A contour detected by CASW is counted as “hit” whenever the contour covers 50% or more of the ground-truth pixels. Since we discard contours that are less than 50 pixels, this means that at least 25 ground-truth pixels need to be detected within the bounding box. Our accuracy in the second clustering stage depends on the initial set of pairs of matching contours (i.e., nodes of graph  $G$ ) input to CASW. This is evaluated by plotting the ROC curve, parameterized by a threshold on the minimum DTW similarity between pairs of matching contours which are included in  $G$ .

**Evaluation in S1:** We present three experiments in S1. *Experiment 1 in S1:* We evaluate the merit of: (a) using pairs of contours as nodes of  $G$ , and (b) accounting for



**Fig. 5.** Evaluation in S1 on the ETHZ dataset. (a): We evaluate five distinct formulations of object discovery, explained in the text, by computing False Positive Rate (FPR) at Bounding Box Hit Rate BBHR=0.5. Our approach  $G$ +CASW gives the best performance. (b): *Precision* and *Recall* as a function of the number of positive examples in the input image set. Performance increases with more positive examples, until about 20 positive images. (c): Evolution of  $\log(p(\mathcal{M})p(G|\mathcal{M}))$  estimated by our CASW (magenta), and standard SW [12] (cyan) on all positive examples of class *Giraffes*, and the same number of negative examples from ETHZ.

spatial configuration of contours as edge weights of  $G$ , against the more common use of individual contours as graph nodes, and contour similarities as edge weights. To this end, we build three weighted graphs  $G_1$ ,  $G_2$  and  $G_3$  of contours extracted only from all positive examples of a single object category in the ETHZ dataset (i.e., the set of negative examples is empty). Nodes of  $G_1$  are individual contours, edges connect candidate matches  $(u, u')$ , and edge weights  $s_{uu'}$  represent the DTW similarity of contours  $u$  and  $u'$ . In  $G_2$  and  $G_3$ , nodes are instead pairs of contours  $(u, u')$ . In  $G_2$ , each edge  $((u, u'), (v, v'))$  receives weight  $(s_{uu'} + s_{vv'})/2$ . In  $G_3$ , edges can only be positive and receive weights  $\rho_e^+$ , defined in Sec. 3. For all three graphs, we apply the standard PageRank algorithm, also used in [9, 17, 10], to identify the most relevant contours, which are then interpreted as object contours. False Positive Rate (FPR) is computed for BBHR = 0.5, and averaged across all categories in the ETHZ dataset. Fig. 5(a) shows that  $G_2$ +PageRank decreases the FPR of  $G_1$ +PageRank by 3.2%. However,  $G_2$ +PageRank still yields a relatively high value of FPR, which suggests that accounting only for shape similarity and ignoring the spatial layout of contours may not be sufficient to handle the very difficult problem of object discovery. Using  $G_3$ +PageRank significantly decreases FPR, which motivates our approach. We also run our CASW on graph  $G_3$ , and on  $G$ , specified in Sec. 3. In comparison with  $G_3$ +CASW, our approach  $G$ +CASW additionally allows the negative polarity of graph edges. Fig. 5(a) shows that  $G_3$ +CASW outperforms  $G_3$ +PageRank, and that  $G$ +CASW gives the best results.

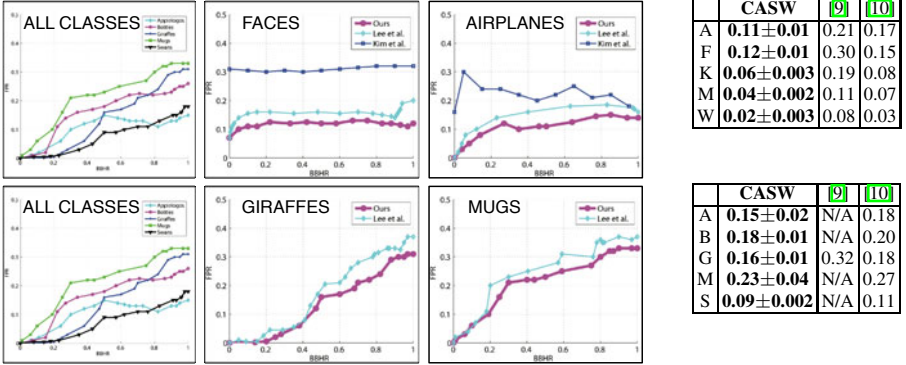
*Experiment 2 in S1:* We test performance in object detection as a function of the number of positive examples in the input image set. The total number of images  $M = 32$  is set to the number of images of the “smallest” class in the ETHZ dataset. In Fig 5(b), we plot the ROC curves when the number of positive images increases, while the number of negative ones proportionally decreases. As expected, performance improves with the increase of positive examples, until reaching a certain number (on average about 20 for the ETHZ dataset).

*Experiment 3 in S1:* Finally, we test our learning of pdf parameters. Fig 5(c) shows the evolution of  $\log(p(\mathcal{M})p(G|\mathcal{M}))$  in the first stage of object discovery in the image set

**Table 2.** Mean purity of category discovery for Caltech-101 (A: Airplanes, C: Cars, F: Faces, M: Motorbikes, W: Watches, K: Ketches), and ETHZ dataset (A: Applelogos, B: Bottles, G: Giraffes, M: Mugs, S: Swans)

Caltech categories	Our method	[10]	[9]	[17]
A,C,F,M	<b>98.62±0.51</b>	98.03	98.55	88.82
A,C,F,M,W	<b>97.57±0.46</b>	96.92	97.30	N/A
A,C,F,M,W,K	<b>97.13±0.42</b>	96.15	95.42	N/A

ETHZ categories	Our method	[10]
A,B,G,M,S (bbox)	<b>96.16±0.41</b>	95.85
A,B,G,M,S (expanded)	<b>87.35±0.37</b>	76.47
A,B,G,M,S (entire image)	<b>85.49±0.33</b>	N/A



**Fig. 6. Bounding Box Hit Rates (BBHR) vs False Positive Rates (FPR).** Top is Caltech-101, bottom is ETHZ. Left column is our CASW on all classes, and middle and right columns show a comparison with [9, 10] on a specific class (lower curves are better). The tables show FPR at BBHR=0.5. Caltech-101: A: Airplanes, F: Faces, K: Ketches, M: Motorbikes, W: Watches. ETHZ: A: Applelogos, B: Bottles, G: Giraffes, M: Mugs, S: Swans. (best viewed in color)

consisting of all positive examples of class *Giraffes*, and the same number of negative examples showing other object categories from the ETHZ dataset. We compare our CASW with the standard SW of [12], where the pdf parameters are not learned, but pre-specified. Since these parameters are unknown, to compute both the ground-truth value and the value produced by [12] of  $\log(p(\mathcal{M})p(G|\mathcal{M}))$ , we use the pdf parameters learned by our approach after CASW converged. As CASW and SW make progress through iterative clustering of the images, Fig. 5(c) shows that CASW yields a steeper increase in  $\log(p(\mathcal{M})p(G|\mathcal{M}))$  to higher values, closer to the ground-truth. Notice that CASW avoids local minima and converges after only few iterations.

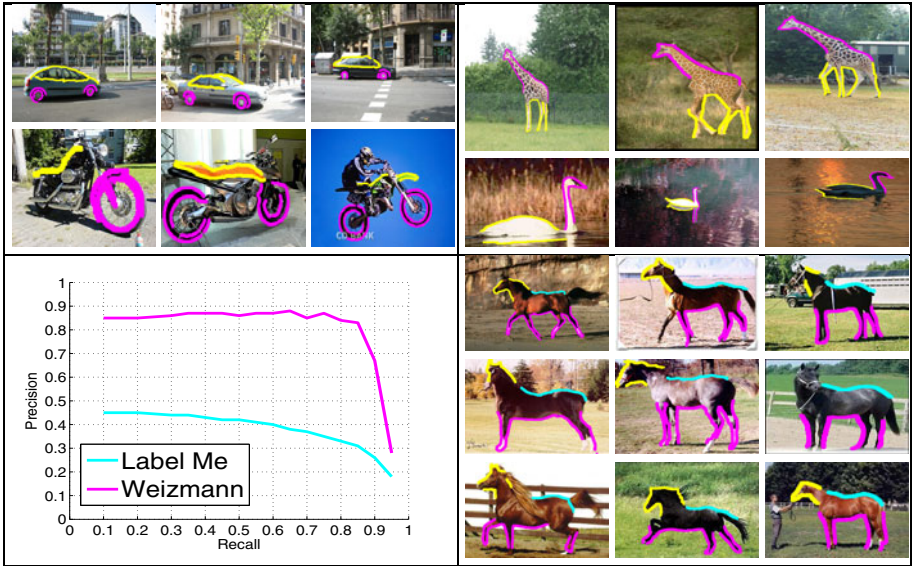
**Evaluation in S2:** We evaluate the first and second stages of object discovery in S2. *First Stage in S2:* We build a graph whose nodes represent entire images. Edges between images in the graph are characterized by weights, defined as an average of DTW similarities of contour matches from the corresponding pair of images. A similar characterization of graph edges is used in [9, 10]. For object discovery, we apply CASW to the graph, resulting in image clusters. Each cluster is taken to consist of images showing a unique object category. Unlike [9, 10], we do not have to specify the number of categories present in the image set, as an input parameter, since it is automatically inferred by CASW. Evaluation is done on Caltech-101 and the ETHZ dataset. Table 2 shows



that our mean purity is superior to that of [9, 17, 10]. On Caltech-101, CASW successively finds  $K = 4, 5, 6$  clusters of images, as we gradually increase the true number of categories from 4 to 6. This demonstrates that we are able to automatically find the number of categories present, with no supervision. On ETHZ, CASW again correctly finds  $K = 5$  categories. As in [10], we evaluate purity when similarity between the images (i.e., weights of edges in the graph) is estimated based on contours falling within: (a) the bounding boxes of target objects, (b) twice the size of the original bounding boxes (called expanded in Table 2), and (c) the entire images. On ETHZ, CASW does not suffer a major performance degradation when moving from the bounding boxes, to the challenging case of using all contours from the entire images. Overall, our purity rates are high, which enables accurate clustering of contours in the second stage.

*Second Stage in S2:* We use contours from all images grouped within one cluster in the first stage to build our graph  $G$ , and then conduct CASW. This is repeated for all image clusters. The clustering of contours by CASW amounts to foreground detection, since the identified contour clusters are taken to represent parts of the discovered object category. We evaluate BBHR and FPR on Caltech-101, ETHZ, LabelMe, and Weizmann Horses. Fig 6 shows that our BBHR and FPR values are higher than those of [9, 10] on the Caltech and ETHZ. CASW finds  $K = 1$  for *Airplanes, Cars Rear, Faces, Ketches, Watches* in Caltech-101, *Apples, Bottles, Mugs* in ETHZ, and *Car rear, Face, Airplane* in LabelMe. These objects do not have articulated parts that move independently, hence, only one contour cluster is found. On the other hand, it finds  $K = 2$  for *Giraffes, Swans* in ETHZ, *Cars side, Motorbikes* in Caltech and LabelMe, and  $K = 3$  for Weizmann Horses. In Fig 7 we highlight contours from different clusters with distinct colors. Fig 7 demonstrates that CASW is capable not only to discover foreground objects, but also to detect their characteristic parts, e.g., wheels and roof for *Cars side*, wheels and seat for *Motorbikes*, head and legs for *Giraffes*, etc. The plot in Fig 7 evaluates our object detection on LabelMe and Weizmann Horses. Detection accuracy is estimated as the standard ratio of intersection over union of ground-truth and detection bounding boxes,  $(BB_{gt} \cap BB_d)/(BB_{gt} \cup BB_d)$ , where  $BB_d$  is the smallest bounding box that encloses detected contours in the image. The average detection accuracy for each category is: [Face(F): 0.52, Airplane(A): 0.45, Motorbike(M): 0.42, Car Rear(C): 0.34], whereas [10] achieves only [(F): 0.48, (A): 0.43, (M): 0.38, (C): 0.31]. For Weizmann Horses, we obtain *Precision* and *Recall* of  $84.9\% \pm 0.68\%$  and  $82.4\% \pm 0.51\%$ , whereas [8] achieves only 81.5% and 78.6%.

**Remark:** The probability of contour patterns that repeat in the background increases with the number of images. On large datasets, CASW is likely to extract clusters of those background patterns. However, the number of contours in these clusters is relatively small, as compared to clusters that contain true object contours, because the frequency of such patterns is, by definition, smaller than that of foreground objects. Therefore, these spurious clusters can be easily identified, and interpreted as background. For example, in setting S1, when the input image set consists of only positive, 100 images of Weizmann Horses, we obtain  $K = 3$  very large clusters (Fig 7), and 9 additional clusters with only 5 to 10 background contours.



**Fig. 7.** Unsupervised detection and segmentation of objects in example images from LabelMe (top left), ETHZ (top right), and Weizmann Horses (bottom right). For LabelMe and ETHZ, each row shows images that are grouped within a unique image cluster by CASW in the first stage. Contours that are clustered by CASW in the second stage are highlighted with distinct colors indicating cluster membership. CASW accurately discovers foreground objects, and delineates their characteristic parts. E.g., for LabeMe *Cars sideview* CASW discovers two contour clusters (yellow and magenta), corresponding to the two car parts wheels and roof. (bottom left) ROC curves for LabelMe and Weizmann Horses, obtained by varying the minimum allowed DTW similarity between pairs of matching contours which are input to CASW. (best viewed in color)

**Implementation.** The C-implementation of our CASW runs in less than 2 minutes on any dataset of less than 100 images, on a 2.40GHz PC with 3.48GB RAM.

## 6 Conclusion

We have shown that shape alone is sufficiently discriminative and expressive to provide robust and efficient object discovery in unlabeled images, without using any photometric features. This is done by clustering image contours based on their intrinsic geometric properties, and spatial layouts. We have also made contributions to the popular research topic in vision, that of probabilistic multicoloring of a graph, including: (a) the on-line learning of pdf parameters governing multicoloring assignments; (b) enforcing stronger positive and negative coupling nodes in the graph, by allowing the polarity of graph edges to dynamically vary during the Metropolis-Hastings (MH) jumps; and (c) regularizing the posterior of multicoloring assignments to help MH jumps escape from trivial solutions. These extensions lead to faster convergence to higher values of the graph's posterior distribution than the well-known SW cut.

## References

1. Biederman, I.: Surface versus edge-based determinants of visual recognition. *Cognitive Psychology* 20, 38–64 (1988)
2. Shotton, J., Blake, A., Cipolla, R.: Multiscale categorical object recognition using contour fragments. *PAMI* 30, 1270–1281 (2008)
3. 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. 3953, pp. 14–28. Springer, Heidelberg (2006)
4. Zhu, Q., Wang, L., Wu, Y., Shi, J.: Contour context selection for object detection: A set-to-set contour matching approach. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008*, Part II. LNCS, vol. 5303, pp. 774–787. Springer, Heidelberg (2008)
5. Kokkinos, I., Yuille, A.L.: HOP: Hierarchical object parsing. In: *CVPR* (2009)
6. Bai, X., Wang, X., Liu, W., Latecki, L.J., Tu, Z.: Active skeleton for non-rigid object detection. In: *ICCV* (2009)
7. Russell, B., Freeman, W., Efros, A., Sivic, J., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: *CVPR* (2006)
8. Todorovic, S., Ahuja, N.: Unsupervised category modeling, recognition, and segmentation in images. *IEEE TPAMI* 30, 1–17 (2008)
9. Kim, G., Faloutsos, C., Hebert, M.: Unsupervised modeling of object categories using link analysis techniques. In: *CVPR* (2008)
10. Lee, Y.J., Grauman, K.: Shape discovery from unlabeled image collections. In: *CVPR* (2009)
11. Felzenszwalb, P., McAllester, D.: A min-cover approach for finding salient curves. In: *CVPR POCV* (2006)
12. Lin, L., Zeng, K., Liu, X., Zhu, S.C.: Layered graph matching by composite cluster sampling with collaborative and competitive interactions. In: *CVPR* (2009)
13. Barbu, A., Zhu, S.C.: Generalizing Swendsen-Wang to sampling arbitrary posterior probabilities. *IEEE TPAMI* 27, 1239–1253 (2005)
14. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. *PAMI* 24, 509–522 (2002)
15. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *IJCV* 60, 91–110 (2004)
16. Chong, E.K.P., Zak, S.H.: *An introduction to optimization*. Wiley-Interscience, Hoboken (2001)
17. Lee, Y.J., Grauman, K.: Foreground focus: Unsupervised learning from partially matching images. In: *BMVC* (2008)
18. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: *CVPR* (2004)
19. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: a database and web-based tool for image annotation. Technical Report AIM-2005-025, MIT (2005)
20. Borenstein, E., Ullman, S.: Class-specific, top-down segmentation. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*, Part II. LNCS, vol. 2351, pp. 109–122. Springer, Heidelberg (2002)

# What Does Classifying More Than 10,000 Image Categories Tell Us?

Jia Deng<sup>1,3</sup>, Alexander C. Berg<sup>2</sup>, Kai Li<sup>1</sup>, and Li Fei-Fei<sup>3</sup>

<sup>1</sup> Princeton University

<sup>2</sup> Columbia University

<sup>3</sup> Stanford University

**Abstract.** Image classification is a critical task for both humans and computers. One of the challenges lies in the large scale of the semantic space. In particular, humans can recognize tens of thousands of object classes and scenes. No computer vision algorithm today has been tested at this scale. This paper presents a study of large scale categorization including a series of challenging experiments on classification with more than 10,000 image classes. We find that a) computational issues become crucial in algorithm design; b) conventional wisdom from a couple of hundred image categories on relative performance of different classifiers does not necessarily hold when the number of categories increases; c) there is a surprisingly strong relationship between the structure of WordNet (developed for studying language) and the difficulty of visual categorization; d) classification can be improved by exploiting the semantic hierarchy. Toward the future goal of developing automatic vision algorithms to recognize tens of thousands or even millions of image categories, we make a series of observations and arguments about dataset scale, category density, and image hierarchy.

## 1 Introduction

Recognizing categories of objects and scenes is a fundamental human ability and an important, yet elusive, goal for computer vision research. One of the major challenges is the sheer scale of the problem, both in terms of the very high dimensional physical space of images, and the large semantic space humans use to describe visual stimuli. In particular, psychologists have postulated that humans are able to categorize at least tens of thousands of objects and scenes [1].

The breadth of the semantic space has important implications. For many real world vision applications, the ability to handle a large number of object classes becomes a minimum requirement, *e.g.* an image search engine or an automatic photo annotator is significantly less useful if it is unable to cover a wide range of object classes. Even for tasks in restricted domains, *e.g.* car detection, to be effective in the real world, an algorithm needs to discriminate against a large number of distractor object categories.

Recent progress on image categorization has been impressive and has introduced a range of features, models, classifiers, and frameworks [2,3,4,5,6,7,8,9,10]. In this paper we explore scaling up the number of categories considered in recognition experiments from hundreds to over 10 thousand in order to move toward

reducing the gap between machine performance and human abilities. Note that this is not simply a matter of training more and more classifiers (although that is a challenging task on its own). With such large numbers of categories there is a concomitant shift in the difficulty of discriminating between them as the categories sample the semantic space more densely. The previously unexplored scale of the experiments in this paper allow this effect to be measured.

Recognition encompasses a wide range of specific tasks, including classification, detection, viewpoint understanding, segmentation, verification and more. In this paper we focus on category recognition, in particular the task of assigning a single category label to an image that contains one or more instances of a category of object following the work of [11,12,13,14].

We conduct the first empirical study of image categorization at near human scale. Some results are intuitive – discriminating between thousands of categories is in fact more difficult than discriminating between hundreds – but other results reveal structure in the difficulty of recognition that was previously unknown. Our key contributions include:

- The first in-depth study of image classification at such a large scale. Such experiments are technically challenging, and we present a series of techniques to overcome the difficulty. (Sec. 5)
- We show that conventional wisdom obtained from current datasets does not necessarily hold in some cases at a larger scale. For example, the ordering by performance of techniques on hundreds of categories is not preserved on thousands of categories. Thus, we cannot solely rely on experiments on the Caltech [13,14] and PASCAL [12] datasets to predict performance on large classification problems. (Sec. 6)
- We propose a measure of similarity between categories based on WordNet [15] – a hierarchy of concepts developed for studying language. Experiments show a surprisingly strong correlation between this purely linguistic metric and the performance of visual classification algorithms. We also show that the categories used in previous object recognition experiments are relatively sparse – distinguishing them from each other is significantly less difficult than distinguishing many other denser subsets of the 10,000 categories. (Sec. 7)
- Object categories are naturally hierarchical. We propose and evaluate a technique to perform hierarchy aware classification, and show that more informative classification results can be obtained. (Sec. 8)

## 2 Related Work

Much recent work on image classification has converged on bag of visual word models (BoW) [16] based on quantized local descriptors [17,3,4] and support vector machines [3,2] as basic techniques. These are enhanced by multi-scale spatial pyramids (SPM) [4] on BoW or histogram of oriented gradient (HOG) [18,4] features. In the current state-of-the-art, multiple descriptors and kernels are combined using either ad hoc or multiple kernel learning approaches [19,5,20,21]. Work in machine learning supports using winner-takes-all between 1-vs-all

classifiers for the final multi-class classification decision [22]. We choose SPM using BoW because it is a key component of many of the best recognition results [19,5,20,21] and is relatively efficient. Recent work allows fast approximation of the histogram intersection kernel SVM, used for SPM, by a linear SVM on specially encoded SPM features [23]. See Appendix for the modifications necessary to allow even that very efficient solution to scale to very large problems.

There are very few multi-class image datasets with many images for more than 200 categories. One is Tiny Images [6], 32x32 pixel versions of images collected by performing web queries for the nouns in the WordNet [15] hierarchy, without verification of content. The other is ImageNet [24], also collected from web searches for the nouns in WordNet, but containing full images verified by human labelers. To date there have been no recognition results on large numbers of categories published for either dataset<sup>1</sup>. Fergus *et al.* explore semi-supervised learning on 126 hand labeled Tiny Images categories [25] and Wang *et al.* show classification on a maximum of 315 categories (< 5%) [26].

Recent work considering hierarchies for image recognition or categorization [27,28,29,30] has shown impressive improvements in accuracy and efficiency, but has not studied classification minimizing hierarchical cost. Related to classification is the problem of detection, often treated as repeated 1-vs-all classification in sliding windows. In many cases such localization of objects might be useful for improving classification, but even the most efficient of the state of the art techniques [7,20,31] take orders of magnitude more time per image than the ones we consider in this study, and thus cannot be utilized given the scale of our experiments.

### 3 Datasets

The goals of this paper are to study categorization performance on a significantly larger number of categories than the current state of the art, and furthermore to delve deeper toward understanding the factors that affect performance. In order to achieve this, a dataset with a large number of categories spanning a wide range of concepts and containing many images is required. The recently released ImageNet dataset consists of more than 10,000,000 images in over 10,000 categories organized by the WordNet hierarchy [24]. The size and breadth of this data allow us to perform multiple longitudinal probes of the classification problem. Specifically we consider the following datasets:

- **ImageNet10K.** 10184 categories from the Fall 2009 release of ImageNet [32], including both internal and leaf nodes with more than 200 images each (a total of 9 million images).
- **ImageNet7K.** 7404 leaf categories from ImageNet10K. Internal nodes may overlap with their descendants, so we also consider this leaf only subset.
- **ImageNet1K.** 1000 leaf categories randomly sampled from ImageNet7K.

---

<sup>1</sup> Previous work on Tiny Images [6] and ImageNet [24] shows only proof of concept classification on fewer than 50 categories.

- **Rand200{a,b,c}**. Three datasets, each containing 200 randomly selected leaf categories. The categories in Rand200a are sampled from ImageNet1K while Rand200b and Rand200c are sampled directly from ImageNet7K.
- **Ungulate183, Fungus134, Vehicle262**. Three datasets containing all the leaf nodes that are descendants of particular parent nodes in ImageNet10K (named by the parent node and number of leaves).
- **CalNet200**. This dataset serves as a surrogate for the Caltech256 dataset – containing the 200 image categories from Caltech256 that exist in ImageNet.

Note that all datasets have non-overlapping categories except ImageNet10K. Following the convention of the PASCAL VOC Challenge, each category is randomly split 50%-50% into a set of training and test images, with a total of 4.5 million images for training and 4.5 million images for testing. All results are averaged over two runs by swapping training and test, except for ImageNet7K and ImageNet10K due to extremely heavy computational cost. In all cases we provide statistical estimates of the expected variation. The number of training images per category ranges from 200 to 1500, with an average of 450.

## 4 Procedure

The main thrust of this paper is image classification: given an image and  $K$  classes, the task is to select one class label. We employ two evaluation measures:

**Mean accuracy.** The accuracy of each class is the percentage of correct predictions, *i.e.* predictions identical to the ground truth class labels. The mean accuracy is the average accuracy across all classes.

**Mean misclassification cost.** To exploit the hierarchical organization of object classes, we also consider the scenario where it is desirable to have non-uniform misclassification cost. For example, misclassifying “dog” as “cat” might not be penalized as much as misclassifying “dog” as “microwave”. Specifically, for each image  $x_i^{(k)} \in X, i = 1, \dots, m$  from class  $k$ , we consider predictions  $f(x_i^{(k)}) : X \rightarrow \{1, \dots, K\}$ , where  $K$  is the number of classes (*e.g.*  $K = 1000$  for ImageNet1K) and evaluate the cost for class  $k$  as  $L_k = \frac{1}{m} \sum_{i=1}^m C_{f(x_i^{(k)}), k}$ , where  $C$  is a  $K \times K$  cost matrix and  $C_{i,j}$  is the cost of classifying the true class  $j$  as class  $i$ . The mean cost is the average cost across all classes. Evaluation using a cost based on the ImageNet hierarchy is discussed in Sec. 8.

We use the following four algorithms in our evaluation experiments as samples of some major techniques used in object recognition:

- **GIST+NN** Represent each image by a single GIST [33] descriptor (a commonly accepted baseline descriptor for scene classification) and classify using  $k$ -nearest-neighbors (kNN) on L2 distance.
- **BOW+NN** Represent each image by a histogram of SIFT [17] codewords and classify using kNN on L1 distance, as a baseline for BoW NN-based methods.
- **BOW+SVM** Represent each image by a histogram of SIFT codewords, and train and classify using linear SVMs. Each SVM is trained to distinguish one class from the rest. Images are classified by the class with largest score (a 1-vs-all framework). This serves as a baseline for classifier-based algorithms.

- **SPM+SVM** Represent each image by a spatial pyramid of histograms of SIFT codewords [4]. Again a 1-vs-all framework is used, but with approximate histogram intersection kernel SVMs [23,34]. This represents a significant component of many state of the art classifiers [19,5,20,21].

## 5 Computation Matters

Working at the scale of 10,000 categories and 9 million images moves computational considerations to the forefront. Many common approaches become computationally infeasible at such large scale.

As a reference, for this data it takes 1 hour on a 2.66GHz Intel Xeon CPU to train *one* binary linear SVM on bag of visual words histograms (including a minimum amount of parameter search using cross validation), using the extremely efficient LIBLINEAR [34]. In order to perform multi-class classification, one common approach is 1-vs-all, which entails training 10,000 such classifiers – requiring more than 1 CPU year for training and 16 hours for testing. Another approach is 1-vs-1, requiring 50 million pairwise classifiers. Training takes a similar amount of time, but testing takes about 8 years due to the huge number of classifiers. A third alternative is the “single machine” approach, *e.g.* Crammer & Singer [35], which is comparable in training time but is not readily parallelizable. We choose 1-vs-all as it is the only affordable option.

Training **SPM+SVM** is even more challenging. Directly running intersection kernel SVM is impractical because it is at least 100× slower ( 100+ years ) than linear SVM [23]. We use the approximate encoding proposed by Maji & Berg [23] that allows fast training with LIBLINEAR. This reduces the total training time to 6 years. However, even this very efficient approach must be modified because memory becomes a bottleneck<sup>2</sup> – a direct application of the efficient encoding of [23] requires 75GB memory, far exceeding our memory limit (16GB). We reduce it to 12G through a combination of techniques detailed in Appendix A.

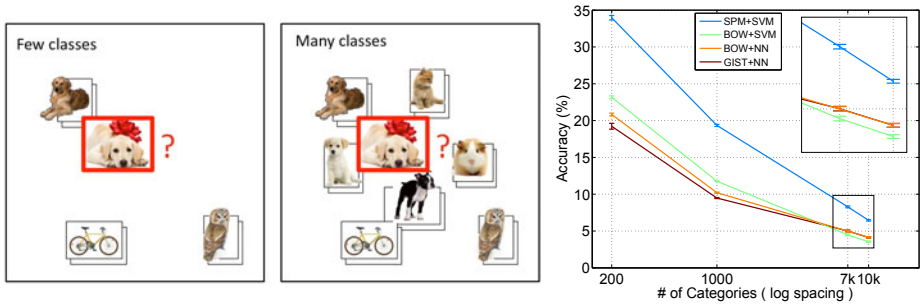
For NN based methods, we use brute force linear scan. It takes 1 year to run through all testing examples for GIST or BOW features. It is possible to use approximation techniques such as locality sensitive hashing [36], but due to the high feature dimensionality (*e.g.* 960 for GIST), we have found relatively small speed-up. Thus we choose linear scan to avoid unnecessary approximation.

In practice, all algorithms are parallelized on a computer cluster of 66 multi-core machines, but it still takes weeks for a single run of all our experiments. Our experience demonstrates that computational issues need to be confronted at the outset of algorithm design when we move toward large scale image classification, otherwise even a baseline evaluation would be infeasible. Our experiments suggest that to tackle massive amount of data, distributed computing and efficient learning will need to be integrated into any vision algorithm or system geared toward real-world large scale image classification.

---

<sup>2</sup> While it is possible to use online methods, *e.g.* stochastic subgradient descent, they can be slower to converge [34].





**Fig. 1.** Given a query image, the task of “image classification” is to assign it to one of the classes (represented by a stack of images) that the algorithm has learned. **Left:** Most traditional vision algorithms have been tested on a small number of somewhat distinct categories. **Middle:** Real world image classification problems may involve a much larger number of categories – so large that the categories can no longer be easily separated. **Right:** Mean classification accuracy of various methods on Rand200{ $a, b, c$ }, ImageNet1K, ImageNet7K and ImageNet10K.

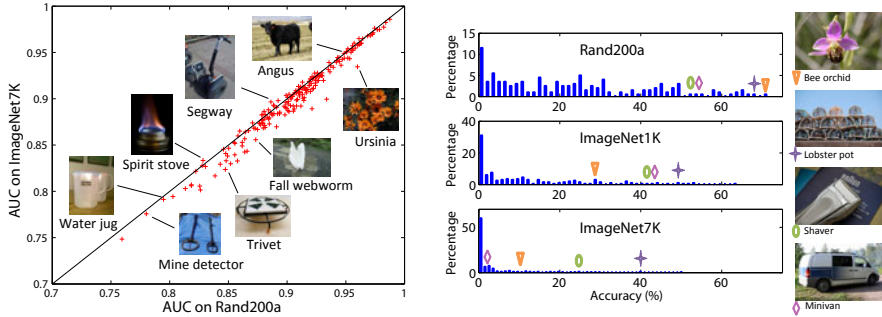
## 6 Size Matters

We first investigate the broad effects on performance and computation of scaling to ten-thousand categories. As the number of categories in a dataset increases, the accuracy of classification algorithms decreases, from a maximum of 34% for Rand200{ $a, b, c$ } to 6.4% for ImageNet10K (Fig. 1 right). While the performance drop comes at no surprise, the speed of decrease is slower than might be expected – roughly a  $2\times$  decrease in accuracy with  $10\times$  increase in the number of classes, significantly better than the  $10\times$  decrease of a random baseline.

There is a surprise from  $k$ -nearest-neighbor ( $kNN$ ) classifiers, either using GIST features or BoW features. For Rand200{ $a, b, c$ }, these techniques are significantly worse than linear classifiers using BoW features, around 10% lower in accuracy. This is consistent with the experience of the field – methods that do use  $kNN$  must be augmented in order to provide competitive performance [2,37]. But the picture is different for ImageNet7K or ImageNet10K categories, where simple  $kNN$  actually outperforms linear SVMs on BoW features (BOW+SVM), with 11-16% higher accuracy. The small absolute gain in mean accuracy, around 0.5%, is made significant by the very small expected standard deviation of the means 0.1%<sup>3</sup>. **A technique that significantly outperforms others on small datasets may actually underperform them on large numbers of categories.**

This apparent breakdown for 1-vs-all with linear classifiers comes despite a consistent line of work promoting this general strategy for multi-class classification [22]. It seems to reveal issues with calibration between classifiers, as the majority of categories have comparable discriminative power on ImageNet7K and Rand200a (Fig 2 left), but multi-way classification is quite poor for ImageNet7K

<sup>3</sup> Stdev for ImageNet7K and ImageNet10K are estimated using the individual category variances, but are very small *cf* standard error and the central limit theorem.



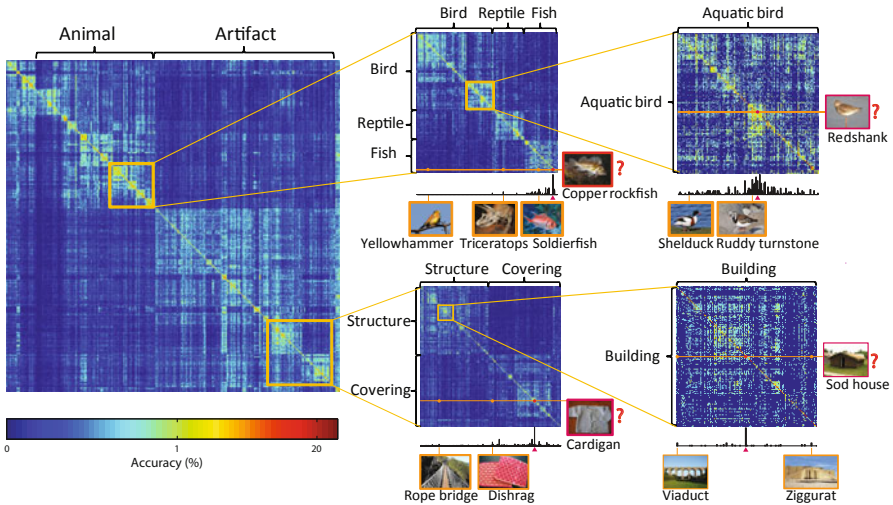
**Fig. 2.** **Left:** Scatter plot comparing the area under ROC curve (AUC) of BOW+SVM for the 200 categories in Rand200a when trained and evaluated against themselves (x-axis) and when trained and evaluated against ImageNet7K (y-axis). **Right:** Histograms of accuracies for the same 200 categories in Rand200a, ImageNet1K, and ImageNet7K, example categories indicated with colored markers.

(Fig 2 right). One explanation is that for the one-against-all approach, a correct prediction would require that the true classifier be more confident than any other classifiers, which becomes more difficult with a larger number of classes as the chance of false alarms from others greatly increases. Then the behavior starts to resemble kNN methods, which are only confident about close neighbors.

Looking in more detail at the confusion between the categories in ImageNet7K reveals additional structure (Fig. 3). Most notable is the generally block diagonal structure, **indicating a correlation between the structure of the semantic hierarchy (by WordNet) and visual confusion between the categories**. The two most apparent blocks roughly align with “artifacts” and “animals”, two very high level nodes in WordNet, suggesting the least amount of confusion between these two classes with more confusion within. This is consistent with both computational studies on smaller numbers of classes [30] and some human abilities [38]. Sections of the confusion matrix are further expanded in Fig. 3. These also show roughly block diagonal structures at increasingly finer levels not available in other datasets. The pattern is roughly block diagonal, but by no means exact. There is a great deal of noise and a fainter “plaid”, oscillating pattern of stripes, indicating that the ordering of categories in WordNet is not completely in agreement with the visual confusion between them.

The block patterns indicate that it is possible to speed up the classification by using a sublinear number of classifiers in a hierarchy, as Griffin & Perona have demonstrated on Caltech256 [30]. They built a hierarchy of classifiers directly from the confusion matrix. Here we confirm their findings by observing a much stronger pattern on a large number of classes. Moreover we note that such a grouping may actually be directly obtained from WordNet, in which case, the output of an internal classifier in the hierarchy would be semantically meaningful.

Also of note is that in scaling to many classes, only a small subset of the distractor classes are truly distracting, possibly explaining the smaller than



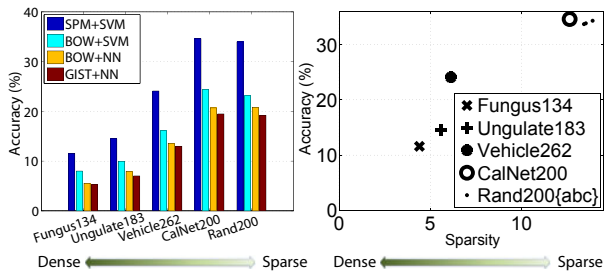
**Fig. 3.** Confusion matrix and sub-matrices of classifying the 7404 leaf categories in ImageNet7K, ordered by a depth first traversal of the WordNet hierarchy, using SPM+SVM. **Left:** Downsampled  $7404 \times 7404$  confusion matrix, each pixel representing max confusion over  $4 \times 4$  entries. **Middle:** Zoom-in to two sub-matrices ( top:  $949 \times 949$ ; bottom:  $1368 \times 1368$  ), each pixel  $2 \times 2$  entries. One row of the matrix is plotted below each matrix (corresponding to red outlined images). The correct class is indicated by a red triangle. Examples of other classes are also shown. **Right:** Further zoom-in (top:  $188 \times 188$ ; bottom:  $145 \times 145$ ), each pixel representing the confusion between two individual categories.

expected performance drop. For example, to classify “German shepherd”, most of the distractor classes are “easy” ones like “dishrag”, while only a few semantically related classes like “husky” add to the difficulty. It suggests that one key to improving large scale classification is to focus on those classes, whose difficulty correlates with semantic relatedness. We quantify this correlation in Sec. 7.

## 7 Density Matters

Our discussion so far has focused on the challenges arising from the sheer number of categories. Figure 3 reveals that the difficulty of recognition varies significantly over different parts of the semantic space. Some classifiers must tackle more semantically related, and possibly visually similar, categories. Accurate classification of such categories leads to useful applications, *e.g.* classifying groceries for assisting the visually impaired, classifying home appliances for housekeeping robots, or classifying insect species for environmental monitoring [39]. We refer to sets of such categories as *dense* and study the effect of density on classification.

We begin by comparing mean classification accuracy for classifiers trained and tested on each of the small datasets – Fungus134, Ungulate183, Vehicle262,



**Fig. 4. Left:** Accuracy on datasets of varying density. Note that CalNet200 (the Caltech 256 categories in ImageNet) has low density and difficulty on par with a set of 200 randomly sampled categories. **Right:** Accuracy (using SPM+SVM) versus dataset density measured by mean distance in WordNet (see Sec. 7).

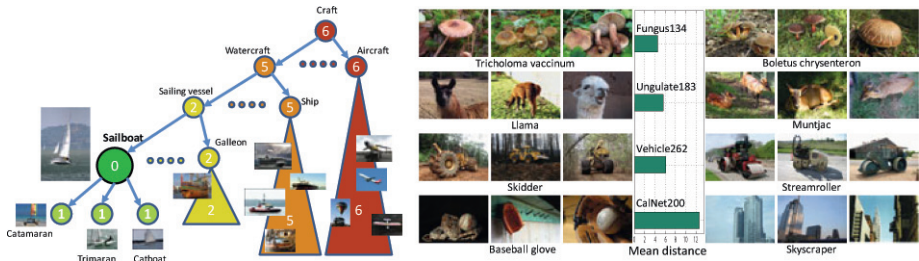
CalNet200, Rand200 – across descriptors and classifiers in Fig. 4. Note that while SPM+SVM produces consistently higher accuracies than the other approaches, the ordering of datasets by performance is exactly the same for each approach. This indicates that **there is a significant difference in difficulty between different datasets, independent of feature and classifier choice.**

Next we try to predict the difficulty of a particular dataset by measuring the density of the categories, based on the hierarchical graph structure of WordNet. We define the distance,  $h(i, j)$ , between categories  $i$  and  $j$ , as the height of their lowest common ancestor. The height of a node is the length of the longest path down to a leaf node (leaf nodes have height 0). We measure the density of a dataset as the mean  $h(i, j)$  between all pairs of categories – smaller implies denser. See Fig. 5 for an illustration and for examples of pairs of categories from each dataset that have distance closest to the mean for that dataset. There is a **very clear correlation between the density in WordNet and accuracy of visual classification; denser datasets predict lower accuracy** (Fig. 4). This is despite the fact that WordNet was not created as a visual hierarchy!

Classification accuracy on 200 randomly chosen categories (Rand200{a,b,c}) is more than 3 times higher than on the 134 categories from Fungus134. The large gap suggests that the methods studied here are not well equipped for classifying dense sets of categories. In fact, there have been relatively few efforts on “dense classification” with some notable exceptions, *e.g.* [40, 41, 39]. The results seem to call for perhaps more specialized features and models, since it is one key to improving large scale classification performance as discussed in Sec. 6.

Also of note is that the Caltech256 categories that occur in ImageNet (CalNet200) have very low density and relatively high accuracy – in almost exactly the same range as random sets of categories. **The Caltech categories are very sparse and do not exhibit the difficulty of dense sets of categories,**

<sup>4</sup> Ordering of datasets is consistent, but ordering of methods may change between datasets as noted in Sec. 6 where BOW+SVM and the kNN approaches switch order.



**Fig. 5. Left:** Illustration of the inter-class distance (indicated by the numbers) between “sailboat” and other classes, as defined in Sec. 7. Any descendant of ship is further from sailboat than galleon but closer than those in aircraft. Note that one step up the hierarchy may increase the distance by more than one as the tree height is the length of the longest path to a leaf node. **Right:** Each row shows a pair of example categories from the dataset indicated in the center column. The pairs are chosen to have distance near the mean distance in WordNet (Sec. 7) between categories in the dataset, indicated by the bars in the center column.

## making Caltech-like datasets incomplete as an evaluation resource towards some of the real-world image classification problems.

Finally we note that our WordNet based measure is not without limitations, *e.g.* “food tuna” and “fish tuna” are semantically related but belong to “food” and “fish” subtrees respectively, so are far away from each other. Nonetheless as a starting point for quantifying semantic density, the results are encouraging.

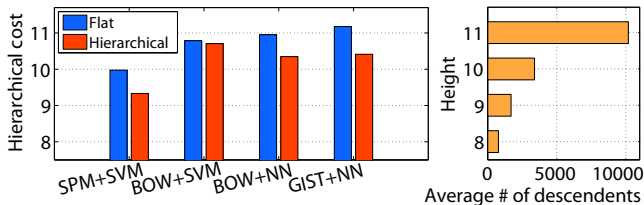
## 8 Hierarchy Matters

For recognition at the scale of human ability, categories will necessarily overlap and display a hierarchical structure [11]. For example, a human may label “redshank” as “shorebird”, “bird”, or “animal”, all of which are correct but with a decreasing amount of information. Humans make mistakes too, but to different degrees at different levels – a “redshank” might be mistaken as a “red-backed sandpiper”, but almost never as anything under “home appliance”.

The implications for real world object classification algorithms are two fold. First a learning algorithm needs to exploit real world data that inevitably has labels at different semantic levels. Second, it is desirable to output labels as informative as possible while minimizing mistakes at higher semantic levels.

Consider an automatic photo annotator. If it cannot classify “redshank” reliably, an answer of “bird” still carries much more information than “microwave”. However, our classifiers so far, trained to minimize the 0-1 loss<sup>5</sup>, have no incentive to do so – predicting “microwave” costs the same as predicting “bird”.

<sup>5</sup> The standard loss function for classification, where a correct classification costs zero and any incorrect classification costs 1.



**Fig. 6. Left:** Hierarchical cost of flat classification and hierarchical classification on ImageNet10K across different methods. **Right:** Mean number of descendants for nodes at each height, indicating the effective log-scale for hierarchical cost.

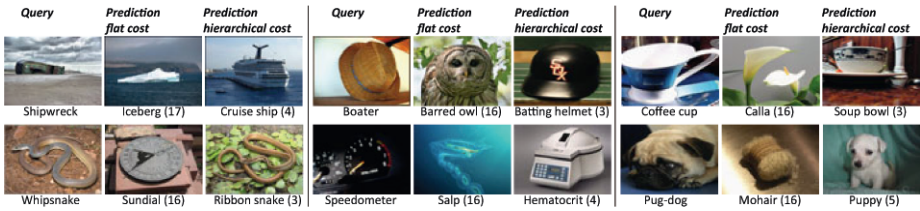
Here we explore ways to make classifiers more informative. We define a hierarchical cost  $C_{i,j}$  for classifying an image of class  $j$  as class  $i$  as  $C_{i,j} = 0$  when  $i = j$  or when  $i$  is a descendant of  $j$ , and  $C_{i,j} = h(i, j)$ , the height of their lowest common ancestor in WordNet, otherwise. This cost definition directly measures the semantic level at which a misclassification occurs – a more informative classifier, one able to discriminate finer details, would have lower cost. It also takes care of the overlapping categories – there is penalty for classifying an image in an internal node as its (more general) ancestor but no cost for classifying it as any of its (more specific) descendants. As an example, in Fig. 5 left, for an image labeled as “sailboat”, classifying it as “catamaran” or any other descendant incurs no cost<sup>6</sup> while classifying as any descendant of “aircraft” incurs cost 6.

We can make various classification approaches cost sensitive by obtaining probability estimates (Appendix). For a query image  $x$ , given posterior probability estimates  $\hat{p}_j(x)$  for class  $j$ ,  $j \in \{1, \dots, K\}$ , according to Bayesian decision theory, the optimal prediction is obtained by predicting the label that minimizes the expected cost  $f(x) = \arg \min_{i=1, \dots, K} \sum_{j=1}^K C_{i,j} \hat{p}_j(x)$ .

Comparing the mean hierarchical cost for the original (flat) classifier with the mean cost for the cost sensitive (hierarchical) classifier, we find a consistent reduction in cost on ImageNet10K (Fig. 6). It shows that the hierarchical classifier can discriminate at more informative semantic levels. While these reductions may seem small, the cost is effectively on a log scale. It is measured by the height in the hierarchy of the lowest common ancestor, and moving up a level can more than double the number of descendants (Fig. 6 right).

The reduction of mean cost on its own would not be interesting without a clear benefit to the results of classification. The examples in Fig. 7 show query images and their assigned class for flat classification and for classification using hierarchical cost. While a whipsnake is misclassified as ribbon snake, it is still correct at the “snake” level, thus giving a more useful answer than “sundial”. It demonstrates that **classification based on hierarchical cost can be significantly more informative.**

<sup>6</sup> The image can in fact be a “trimaran”, in which case it is not entirely correct to predict “catamaran”. This is a limitation of intermediate level ground truth labels.



**Fig. 7.** Example errors using a flat vs. hierarchical classifier with SPM+SVM on ImageNet10K, shown in horizontal groups of three: a query, prediction by a flat classifier (minimizing 0-1 loss), and by a hierarchical classifier (minimizing hierarchical cost). Numbers indicate the hierarchical cost of that misclassification.

## 9 Conclusion

We have presented the first large scale recognition experiments on 10,000+ categories and 9+ million images. We show that challenges arise from the size and density of the semantic space. Surprisingly the ordering of NN and Linear classification approaches swap from previous datasets to our very large scale experiments – we cannot always rely on experiments on small datasets to predict performance at large scale. We produce a measure of category distance based on the WordNet hierarchy and show that it is well correlated with the difficulty of various datasets. We present a hierarchy aware cost function for classification and show that it produces more informative classification results. These experiments point to future research directions for large scale image classification, as well as critical dataset and benchmarking issues for evaluating different algorithms.

**Acknowledgments.** We thank Chris Baldassano, Jia Li, Olga Russakovsky, Hao Su, Bangpeng Yao and anonymous reviewers for their helpful comments. This work is partially supported by an NSF CAREER grant and a Google Award to L.F-F, and by NSF grant 0849512, Intel Research Council and Gigascale Systems Research Center.

## References

1. Biederman, I.: Recognition by components: A theory of human image understanding. *PsychR* 94, 115–147 (1987)
2. Zhang, H., Berg, A.C., Maire, M., Malik, J.: SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition. In: *CVPR 2006* (2006)
3. Grauman, K., Darrell, T.: The pyramid match kernel: Discriminative classification with sets of image features. In: *ICCV* (2005)
4. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: *CVPR 2006* (2006)
5. Varma, M., Ray, D.: Learning the discriminative power-invariance trade-off. In: *ICCV* (2007)
6. Torralba, A., Fergus, R., Freeman, W.: 80 million tiny images: A large data set for nonparametric object and scene recognition. *PAMI* 30, 1958–1970 (2008)

7. Felzenszwalb, P., Mcallester, D., Ramanan, D.: A discriminatively trained, multi-scale, deformable part model. In: CVPR 2008 (2008)
8. Tuytelaars, T., Mikolajczyk, K.: Local Invariant Feature Detectors: A Survey. *Foundations and Trends in Computer Graphics and Vision* 3, 177–820 (2008)
9. Fei-Fei, L., Fergus, R., Torralba, A.: Recognizing and learning object categories. *CVPR Short Course* (2007)
10. Fei-Fei, L., Fergus, R., Torralba, A.: Recognizing and learning object categories. *ICCV Short Course* (2009)
11. Rosch, E., Mervis, C.B., Gray, W.D., Johnson, D.M., Braem, P.B.: Basic objects in natural categories. *Cognitive Psychology* 8, 382–439 (1976)
12. Everingham, M., Zisserman, A., Williams, C.K.I., van Gool, L., et al.: The 2005 pascal visual object classes challenge. In: Quiñero-Candela, J., Dagan, I., Magnini, B., d’Alché-Buc, F. (eds.) *MLCW 2005. LNCS (LNAI)*, vol. 3944, pp. 117–176. Springer, Heidelberg (2006)
13. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. *PAMI* 28, 594–611 (2006)
14. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical Report 7694, Caltech (2007)
15. Fellbaum, C.: *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge (1998)
16. 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)
17. Lowe, D.: Distinctive image features from scale-invariant keypoints. *IJCV* 60, 91–110 (2004)
18. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *CVPR 2005*, pp. 886–893 (2005)
19. Chum, O., Zisserman, A.: An exemplar model for learning object classes. In: *CVPR 2007*, pp. 1–8 (2007)
20. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: *ICCV* (2009)
21. Gehler, P.V., Nowozin, S.: On feature combination for multiclass object classification. In: *ICCV* (2009)
22. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. *JMLR* 5, 101–141 (2004)
23. Maji, S., Berg, A.C.: Max-margin additive models for detection. In: *ICCV* (2009)
24. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: *CVPR 2009* (2009)
25. Fergus, R., Weiss, Y., Torralba, A.: Semi-supervised learning in gigantic image collections. In: *NIPS* (2009)
26. Wang, C., Yan, S., Zhang, H.J.: Large scale natural image classification by sparsity exploration. *ICASP* (2009)
27. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: *CVPR 2006*, pp. II: 2161–2168 (2006)
28. Marszalek, M., Schmid, C.: Semantic hierarchies for visual object recognition. In: *CVPR 2007*, pp. 1–7 (2007)
29. Zweig, A., Weinshall, D.: Exploiting object hierarchy: Combining models from different category levels. In: *ICCV 2007*, pp. 1–8 (2007)
30. Griffin, G., Perona, P.: Learning and using taxonomies for fast visual categorization. In: *CVPR 2008* (2008)



31. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge, VOC 2009 Results (2009), <http://www.pascal-network.org/challenges/VOC/voc2009/workshop/>
32. <http://www.image-net.org>
33. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV* 42, 145–175 (2001)
34. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. *JMLR* 9, 1871–1874 (2008)
35. Crammer, K., Singer, Y., Cristianini, N., Shawe-Taylor, J., Williamson, B.: On the algorithmic implementation of multiclass kernel-based vector machines. *JMLR* 2 (2001)
36. Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In: *FOCS 2006*, pp. 459–468 (2006)
37. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: *CVPR 2008* (2008)
38. Thorpe, S., Fize, D., Marlot, C.: Speed of processing in the human visual system. *Nature* 381, 520–522 (1996)
39. Martinez-Munoz, G., Larios, N., Mortensen, E., Zhang, W., Yamamuro, A., Paasch, R., Payet, N., Lytle, D., Shapiro, L., Todorovic, S., Moldenke, A., Dietterich, T.: Dictionary-free categorization of very similar objects via stacked evidence trees. In: *CVPR 2009* (2009)
40. Nilsback, M.E., Zisserman, A.: A visual vocabulary for flower classification. In: *CVPR 2006*, pp. 1447–1454 (2006)
41. Ferencz, A., Learned-Miller, E.G., Malik, J.: Building a classification cascade for visual identification from one example. In: *ICCV 2005*, pp. 286–293 (2005)
42. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008), <http://www.vlfeat.org>
43. Lin, H.T., Lin, C.J., Weng, R.C.: A note on platt’s probabilistic outputs for support vector machines. *Mach. Learn.* 68, 267–276 (2007)

## A Experimental Details

We obtain BoW histograms (L1-normalized) using dense SIFT [42] on 20x20 overlapping patches with a spacing of 10 pixels at 3 scales on images resized to a max side length of 300, and a 1000 codebook from KMeans on 10 million SIFT vectors. We use the same codewords to obtain spatial pyramid histograms (3 levels),  $\phi_2$  encoded [23] to approximate the intersection kernel with linear SVMs. Due to high dimensionality (21k), we only encode nonzeros (but add a bias term). This preserves the approximation for our, non-negative, data, but with slightly different regularization. We found no empirical performance difference testing up to 1K categories. To save memory, we use only two bytes for each entry of encoded vectors (sparse) by delta-coding its index (1 byte) and quantizing its value to 256 levels (1 byte). We further reduce memory by only storing every other entry, exploiting redundancy in consecutive entries. We use LIBLINEAR [34] to train linear SVMs, parameter C determined by searching over 3 values (0.01, 0.1, 1 for ImageNet10K) with 2-fold cross validation. We use smaller weight for negative examples (100× smaller for ImageNet10K) than positives. We obtain posterior probability estimates by fitting a sigmoid function to the outputs of SVMs [43], or by taking the percent of neighbors from a class for NN.

# Modeling and Analysis of Dynamic Behaviors of Web Image Collections

Gunhee Kim<sup>1</sup>, Eric P. Xing<sup>1</sup>, and Antonio Torralba<sup>2</sup>

<sup>1</sup> Carnegie Mellon University, Pittsburgh, PA 15213, USA

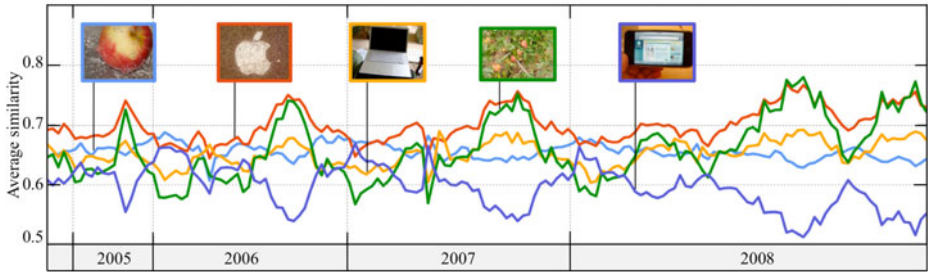
<sup>2</sup> Massachusetts Institute of Technology, Cambridge, MA 02139, USA  
{gunhee,epxing}@cs.cmu.edu, torralba@csail.mit.edu

**Abstract.** *Can we model the temporal evolution of topics in Web image collections? If so, can we exploit the understanding of dynamics to solve novel visual problems or improve recognition performance?* These two challenging questions are the motivation for this work. We propose a nonparametric approach to modeling and analysis of topical evolution in image sets. A scalable and parallelizable sequential Monte Carlo based method is developed to construct the similarity network of a large-scale dataset that provides a base representation for wide ranges of dynamics analysis. In this paper, we provide several experimental results to support the usefulness of image dynamics with the datasets of 47 topics gathered from Flickr. First, we produce some interesting observations such as tracking of subtopic evolution and outbreak detection, which cannot be achieved with conventional image sets. Second, we also present the complementary benefits that the images can introduce over the associated text analysis. Finally, we show that the training using the *temporal association* significantly improves the recognition performance.

## 1 Introduction

This paper investigates the discovery and use of topical evolution in Web image collections. The images on the Web are rapidly growing, and it is obvious to assume that their topical patterns evolve over time. Topics may rise and fall in their popularity; sometimes they are split or merged to a new one; some of them are synchronized or mutually exclusive on the timeline. In Fig. 1, we download *apple* images and their associated timestamps from Flickr, and measure the similarity changes with some canonical images of *apple*'s subtopics. As *Google trends* reveal the popularity variation of query terms in the search volumes, we can easily observe the affinity changes of each subtopic in the *apple* image set.

The main objectives of this work are as follows. First, we propose a nonparametric approach to modeling and analysis of temporal evolution of topics in Web image collections. Second, we show that understanding image dynamics is useful to solve novel problems such as subtopic outbreak detection and to improve classification performance using the *temporal association* that is inspired by studies in human vision [2, 19, 21]. Third, we present that the images can be a



**Fig. 1.** The *Google trends*-like visualization of the subtopic evolution in the *apple* images from Flickr (*fruit*: blue, *logo*: red, *laptop*: orange, *tree*: green, *iphone*: purple). We choose the cluster center image of each subtopic, and measure the average similarity with the posterior (*i.e.* a set of weighted image samples) at each time step. The *fruit* subtopic is stable along the timeline whereas the *iphone* subtopic is highly fluctuated.

more reliable and delicate source of information to detect topical evolution than the texts.

Our approach is motivated by the recent success of the nonparametric methods [13,20] that are powered by large databases. Instead of using sophisticated parametric topic models [3,22], we represent the images with timestamps in the form of a *similarity network* [11], in which vertices are images and edges connect the temporally related and visually similar images. Thus, our approach is able to perform diverse dynamics analysis without solving complex inference problems. For example, a simple information-theoretic measure of the network can be used to detect subtopic outbreaks, which point out when the evolution speed is abruptly changed. The *temporal context* is also easily integrated with the classifier training in a framework of the Metropolis-Hastings algorithm.

The network generation is based on the sequential Monte Carlo (*i.e.* particle filtering) [11,9]. In the sequential Monte Carlo, the posterior (*i.e.* subtopic distribution) at a particular time step is represented by a set of weighted image samples. We track similar subtopics (*i.e.* clusters of images) in consecutive posteriors along the timeline, and create edges between them. The sampling based representation is quite powerful in our context. Since we deal with unordered natural images on the Web, any Gaussian or linearity assumption does not hold and multiple peaks of distributions are unavoidable. Another practical advantage is that we can easily control the tradeoff between accuracy and speed by managing the number of samples and parameters in the transition model. The proposed algorithm is easily parallelizable by running multiple sequential Monte Carlo trackers with different initialization and parameters. Our approach is also scalable and fast. The computation time is linear with the number of images.

For evaluation, we download more than 9M images of 47 topics from Flickr. Most standard datasets in computer vision research [7,18] have not yet considered the importance of temporal context. Recently, several datasets have introduced *spatial contexts* as fundamental cues to recognition [18], but the support for temporal context has still been largely ignored. Our experiments clearly show

that our modeling and analysis is practically useful and can be used to understand and simulate human-like visual experience from Web images.

## 1.1 Related Work

The temporal information is one of the most obvious features in video or auditory applications. Hence, here we review only the use of temporal cues for image analysis. The importance of temporal context has long been recognized in neuroscience research [2,19,21]. Wide range of research has supported that the *temporal association* (*i.e.* liking temporally close images) is an important mechanism to recognize objects and generalize visual representation. [21] tested several interesting experiments to show that temporally correlated multiple views can be easily linked to a single representation. [2] proposed a learning model for 3D object recognition by using the temporal continuity in image sequences.

In computer vision, [16] is one of the early studies that use temporal context in active object recognition. They used a POMDP framework for the modeling of temporal context to disambiguate the object hypotheses. [5] proposed a HMM-based temporal context model to solve scene classification problems. For the indoor-outdoor classification and the sunset detection, they showed that the temporal model outperformed the baseline content-based classifiers.

As the Internet vision emerges as an active research area in computer vision, timing information starts to be used in the assistance of visual tasks. Surprisingly, however, the dynamics or temporal context for Web images has not yet been studied a great deal, contrary to the fact that the study of the dynamic behaviors of the texts on the Web has been one of active research areas in data mining and machine learning communities [3,22]. We briefly review some notable examples using timestamp meta-data for visual tasks. [6] developed an annotation method for personal photo collections, and the timestamps associated with the images were used for better correlation discovery between the images. [12] proposed a landmark classification for an extremely large dataset, and the temporal information was used for the constraints to remove misclassification. [17] also used the timestamp as an additional feature to develop an object and event retrieval system for online image communities. [10] presented a method to geolocate a sequence of images taken by a single individual. Temporal constraints from the sequence of images were used as a strong prior to improve the geolocation accuracy.

The main difference between their work and ours is that they considered the temporal information as additional meta-data or constraints to achieve their original goals (*i.e.* annotations in [6], classification and detection in [12,17], and the geolocation of images in [10]). However, our work considers the timestamps associated with images as a main research subject to uncover dynamic behaviors of Web images. To our best knowledge, there have been very few previous attempts to tackle this issue in computer vision research.

## 2 Network Construction by Sequential Monte Carlo

### 2.1 Image Description and Similarity Measure

Each image is represented by two types of descriptors, which are spatial pyramids of visual words [14] and HOG [4]. We use the codes provided by the authors of the papers. A dictionary of 200 visual words is formed by K-means to randomly selected SIFT descriptors [14]. A visual word is densely assigned to every pixel of an image by finding the nearest cluster center in the dictionary. Then visual words are binned using a two-level spatial pyramid. The oriented gradients are computed by Canny edge detection and Sobel mask [4]. The HOG descriptor is then discretized into 20 orientation bins in the range of  $[0^\circ, 180^\circ]$ . Then the HOG descriptors are binned using a three-level spatial pyramid. The similarity measure between a pair of images is the cosine similarity, which is calculated by the dot product of a pair of  $L_2$  normalized descriptors.

### 2.2 Problem Statement

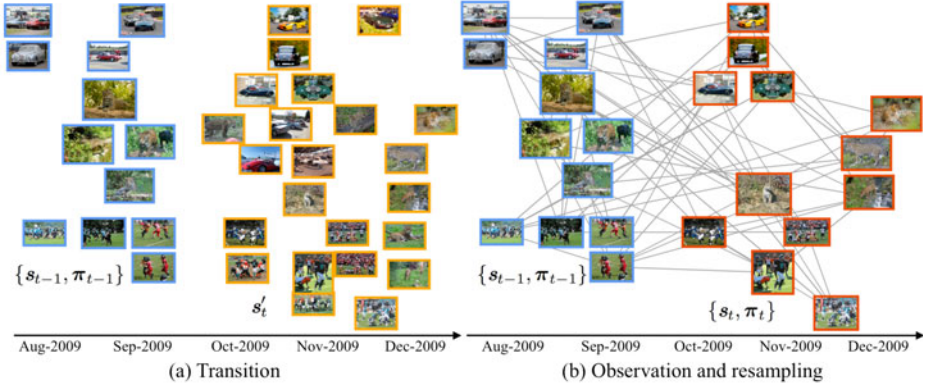
The input of our algorithm is a set of images  $\mathcal{I} = \{I_1, I_2, \dots, I_N\}$  and associated tags of taken time  $\mathcal{T} = \{T_1, T_2, \dots, T_N\}$ . The main goal is to generate an  $N \times N$  sparse similarity network  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$  by using the Sequential Monte Carlo (SMC) method. Each vertex in  $\mathcal{V}$  is an image in the dataset. The edge set  $\mathcal{E}$  is created between the images that are visually similar and temporally distant with a certain interval that is assigned by the *transition model* of the SMC tracker (Section 2.3). The weight set  $\mathcal{W}$  is discovered by the similarity between descriptors of images (Section 2.1). For sparsity, each image is connected to its  $k$ -nearest neighbors with  $k = a \log N$ , where  $a$  is a constant (e.g.  $a = 10$ ).

### 2.3 Network Construction Using Sequential Monte Carlo

Algorithm 1 summarizes the proposed SMC based network construction. For better readability, we follow the notation of *condensation* algorithm [9]. The output of each iteration of the SMC is the conditional subtopic distribution (i.e. posterior) at every step, which is approximated by a set of images with relative importance denoted by  $\{\mathbf{s}_t, \boldsymbol{\pi}_t\} = \{s_t^{(i)}, \pi_t^{(i)}, i = 1, \dots, M\}$ . Note that our SMC does not explicitly solve the *data association* during the tracking. In other words, we do not assign a subtopic membership to each image in  $\mathbf{s}_t$ . However, it can be easily obtained later by applying clustering to the subgraph of  $\mathbf{s}_t$ .

Fig. 2 shows a downsampled example of a single iteration of the posterior estimation. At every iteration, the SMC generates a new posterior  $\{\mathbf{s}_t, \boldsymbol{\pi}_t\}$  by running *transition*, *observation*, and *resampling*.

The image data are severely unbalanced on the timeline. (e.g. There are only a few images within a month in 2005 but a large number of images within even a week in 2008). Thus, in our experiments, we bin the timeline by the number of images instead of a fixed time interval. (e.g. The timeline may be binned by every 3000 images instead of a month). The function  $\tau(T_i, m)$  is used to indicate the timestamp of the  $m$ -th image later from the image at  $T_i$ .



**Fig. 2.** An overview of the SMC based network construction for the *jaguar* topic. The subtopic distribution at each time step is represented by a set of weighted image samples (*i.e.* posterior)  $\{s_t, \pi_t\}$ . In this example, a posterior of the *jaguar* topic consists of image samples of *animal*, *cars*, and *football* subtopics. (a) The transition model generates new posterior candidates  $s'_t$  from  $s_{t-1}$ . (b) The observation model discovers  $\pi'_t$  of  $s'_t$  and the resampling step computes  $\{s_t, \pi_t\}$  from  $\{s'_t, \pi'_t\}$ . Finally, the network is constructed by similarity matching between two consecutive posteriors  $s_{t-1}$  and  $s_t$ .

**Initialization.** The initialization samples the initial posterior  $s_0$  from the prior  $p(x_0)$  at  $T_0$ .  $p(x_0)$  is set by a Gaussian distribution  $N(T_0, \tau^2(T_0, 2M/3))$  on the timeline, which means that  $2M$  numbers of images around  $T_0$  have nonzero probabilities to be selected as one of  $s_0$ . The initial  $\pi_0$  is uniformly set to  $1/M$ .

**Transition Model.** The transition model generates posterior candidates  $s'_t$  rightward on the timeline from the previous  $\{s_{t-1}, \pi_{t-1}\}$  (See Fig. 2 (a) for an example). Each image  $s_{t-1}^{(i)}$  in  $s_{t-1}$  recommends  $m_i$  numbers of images that are similar to itself as candidates set  $s'_t$  for the next posterior. A more weighted image  $s_{t-1}^{(i)}$  is able to recommend more images for  $s'_t$ . ( $\sum_i m_i = 2M$  and  $m_i \propto \pi_{t-1}^{(i)}$ ). At this stage, we generate  $2M$  candidates (*i.e.*  $|s'_t| = 2M$ ), and the observation and resampling steps reduce it to be  $|s_t| = M$  while computing weights  $\pi_t$ .

Similarly to condensation algorithm [9], the transition consists of deterministic *drift* and stochastic *diffusion*. The *drift* describes the transition tendency of the overall  $s'_t$  (*i.e.* how far the  $s'_t$  is located from the  $s_{t-1}$  on the timeline). The *diffusion* assigns a random transition of an individual image. The *drift* and the *diffusion* are modeled by a Gaussian distribution  $N(\mu_t, \sigma^2)$  and a Gamma distribution  $\Gamma(\alpha, \beta)$ , respectively. The final transition model is the product of these two distributions [8] in Eq. 1. The asterisk of  $P_t^{(i)*}(x)$  in Eq. 1 means that it is not normalized. Renormalization is not required since we will use *importance sampling* to sample images on the timeline with the target distribution (See the next subsection with Fig. 3 for the detail).

$$P_t^{(i)*}(x) = N(x; \mu_t, \sigma^2) \times \Gamma(x; \alpha_{t-1}^{(i)}, \beta_{t-1}^{(i)}) \quad (1)$$

---

**Algorithm 1.** The SMC based network generation

---

**Input:** (1) A set of images  $\mathcal{I}$  sorted by timestamps  $\mathcal{T}$ . (2) Start time  $T_0$  and end time  $T_e$ . (3) Posterior size  $M$ . (4) Parameters for *drift*:  $(\Delta M_\mu, \sigma^2)$ .

**Output:** Network  $G$

**Initialization:**

1: draw  $s_0^{(i)} \sim N(T_0, \tau^2(T_0, 2M/3))$ ,  $\pi_0^{(i)} = 1/M$  for  $i = 1, \dots, M$ .

**while**  $\mu_t < T_e$ , ( $\mu_0 = T_0$  and  $\mu_t = \mu_{t-1} + \tau(\mu_{t-1}, \Delta M_\mu)$ ). **do**

## [Transition]

**for all**  $s_{t-1}^{(i)} \in s_{t-1}$  with  $x^{(i)} = \emptyset$  **do**

**repeat**

3: draw  $x \sim N(x; \mu_t, \sigma^2) \times \Gamma(x; \alpha_{t-1}^{(i)}, \beta_{t-1}^{(i)})$  ( $\alpha_{t-1}^{(i)} \propto 1/\pi_{t-1}^{(i)}$ ,  $\beta_{t-1}^{(i)} = \mu_t/\alpha_{t-1}^{(i)}$ ).

4:  $x^{(i)} \leftarrow x$  with probability of  $w(s_{t-1}^{(i)}, x)$ .

**until**  $|x^{(i)}| = m_i = 2M \times \pi_{t-1}^{(i)}$ . Then,  $s'_t \leftarrow x^{(i)}$ .

**end for**

## [Observation]

4: Compute self-similarity graph  $W_t$  of  $s'_t$ . Row-normalize  $W_t$  to  $\widetilde{W}_t$ .

5: Compute the stationary distribution  $\pi'_t$  by solving  $\pi'_t = \widetilde{W}_t^T \pi'_t$ .

## [Resampling]

6: Resample  $\{s_t, \pi_t\}_{i=1}^M$  from  $\{s'_t, \pi'_t\}$  by *systematic sampling* and normalize  $\pi_t$ .

7:  $G \leftarrow W_t(s_t, s_t)$ ,  $W_{t-1,t}(s_{t-1}, s_t)$ , and then convert  $G$  into a  $k$ -NN graph.

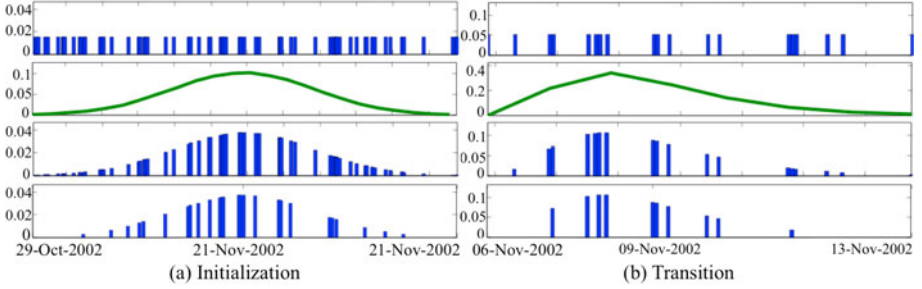
**end while**

---

In sum, for each  $s_{t-1}^{(i)}$ , we sample an image  $x$  using the distribution of Eq. 1, which constrains the position of  $x$  on the timeline. In addition,  $x$  is required to be visually similar to its recommender. Thus, the sample  $x$  is accepted with probability of  $w(s_{t-1}^{(i)}, x)$ , which is the cosine similarity between the descriptors of  $s_{t-1}^{(i)}$  and  $x$ . This process is repeated until  $m_i$  number of samples are accepted.

In Eq. 1, the mean  $\mu_t$  of  $N(\mu_t, \sigma^2)$  is updated at every step as  $\mu_t = \mu_{t-1} + \tau(\mu_{t-1}, \Delta M_\mu)$  where  $\Delta M_\mu$  is the control parameter for the speed of the tracking. The higher  $\Delta M_\mu$ , the further  $s_t$  is located from  $s_{t-1}$  and the fewer the steps are executed until completion. The variance  $\sigma^2$  of  $N(\mu_t, \sigma^2)$  controls the spread of  $s_t$  along the timeline. A higher  $\sigma^2$  results in a  $s_t$  that includes images with a longer time range.

A Gamma distribution  $\Gamma(\alpha, \beta)$  is usually used to model the time required for  $\alpha$  occurrences of events that follow a Poisson process with a constant rate  $\beta$ . In our interpretation, given an image stream, we assume that the occurrence of images of each subtopic follows the Poisson process with  $\beta$ . Then,  $\Gamma(\alpha_{t-1}^{(i)}, \beta_{t-1}^{(i)})$  of Eq. 1 indicates the time required for the next  $\alpha$  images that have the same subtopic with  $s_{t-1}^{(i)}$  in the image stream. Based on this intuition, the  $\alpha_{t-1}^{(i)}$  for each  $s_{t-1}^{(i)}$  is adjustively selected. A smaller  $\alpha_{t-1}^{(i)}$  is chosen for the image  $s_{t-1}^{(i)}$  with higher  $\pi_{t-1}^{(i)}$  since the similar images to a more weighted  $s_{t-1}^{(i)}$  are likely to occur more frequently in the dataset. The mean of Gamma distribution of each  $s_{t-1}^{(i)}$  is aligned with the mean of the sample set  $\mu_t$ . Therefore,  $\beta_{t-1}^{(i)} = \mu_t/\alpha_{t-1}^{(i)}$  since the mean of Gamma is  $\alpha/\beta$ .



**Fig. 3.** An example of sampling images on the timeline during (a) the initialization and (b) the transition. From top to bottom: The first row shows the image distributions along the timeline. The images are regarded as the samples  $(\{x^{(r)}\}_{r=1}^R)$  from a proposal distribution  $Q^*(x)$ . They are equally weighted (*i.e.*  $Q^*(x^{(r)}) = 1$ ). The second row shows the target distribution  $P^*(x)$ . (*e.g.* Gaussian in (a) and the product of Gaussian and Gamma in (b)). The third row shows the image samples weighted by  $P^*(x^{(r)})/Q^*(x^{(r)})$ . The fourth row shows the images chosen by *systematic sampling* [1].

The main reason to adopt the *product model* rather than the *mixture model* in Eq. 1 is as follows. The *product model* only has a meaningful probability for an event when none of its component distribution has a low probability. (*i.e.* if one of two distributions has zero probability, their product does as well). It is useful in our application that the product with the Gaussian of the *drift* prevents the sampled images from severely spreading along the timeline by setting almost zero probability for the image outside the  $3\sigma$  from  $\mu_t$ .

**Sampling Images with Target Distribution.** In the initialization and the transition, we sample a set of images on the timeline from a given target distribution  $P^*(x)$ . (*e.g.* Gaussian in the initialization and the product of Gaussian and Gamma in the transition). Fig. 3 shows our sampling method, which can be viewed as an *importance sampling* [15]. The importance sampling is particularly useful for the transition model since there is no closed form of the product of Gaussian and Gamma distributions and its normalization is not straightforward.

**Observation Model.** The goal of the observation model is to generate weights  $\pi'_t$  for the  $s'_t$ . First, the similarity matrix  $\mathbf{W}_t$  of  $s'_t$  is obtained by computing pairwise cosine similarity of  $s'_t$ . The  $\pi'_t$  is the stationary distribution of  $\mathbf{W}_t$  by solving  $\pi'_t = \widetilde{\mathbf{W}}_t^T \pi'_t$  where  $\widetilde{\mathbf{W}}_t$  is row-normalized from  $\mathbf{W}_t$  so that  $\widetilde{w}_{ij} = w_{ij} / \sum_k w_{ik}$ .

**Resampling.** The final posterior  $\{s_t, \pi_t\} = \{s_t^{(i)}, \pi_t^{(i)}\}_{i=1}^M$  is resampled from  $\{s'_t, \pi'_t\}$  by running the *systematic sampling* [1] on  $\pi''_t$ . Then  $\pi_t$  is normalized so that their sum is one. The network  $\mathbf{G}$  stores  $\mathbf{W}_t(s_t, s_t)$  and the similarity matrix  $\mathbf{W}_{t-1,t}(s_{t-1}, s_t)$  between two consecutive posteriors  $s_{t-1}$  and  $s_t$ . As discussed in section 2.2, each vertex in  $\mathbf{G}$  is connected to only its  $k$ -nearest neighbors.



### 3 Analysis and Results

#### 3.1 Flickr Dataset

Table 1 summarizes 47 topics of our Flickr dataset. The topic name is identical to the query word. We downloaded all the images containing the query word. They are the images shown when a query word is typed in Flickr’s search box without any option change. For the timestamp, we use the *date\_taken* field of each image that Flickr provides.

We generate the similarity network of each topic by using the proposed SMC based tracking. The runtime is  $O(NM)$  where  $M$  is constant and  $M \ll N$  (i.e.  $1000 \leq M \leq 5000$  in our experiments). The network construction is so fast that, for example, it took about 4 hours for the *soccer* topic with  $N = 1.1 \times 10^6$  and  $M = 5,000$  in a matlab implementation on a single PC. The analysis of the network is also fast since most network analysis algorithms depend on the number of nonzero elements, which is  $O(N \log N)$ .

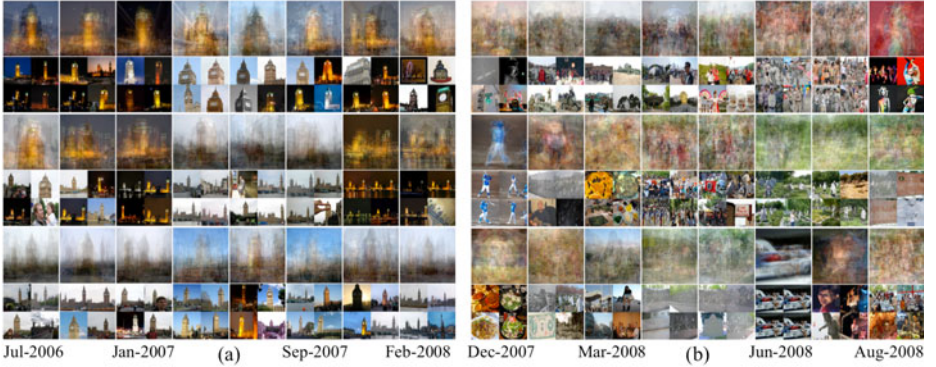
#### 3.2 Evolution of Subtopics

Fig. 4 shows the examples of the subtopic evolution of two topics, *big+ben* and *korean*. As we discussed in previous section, the SMC tracker generates the posterior sets  $\{s_0, \dots, s_e\}$ . Five clusters in each posterior are discovered by applying spectral clustering to the subgraph  $G_t$  of each  $s_t$  in an unsupervised way. Obviously, the dynamic behavior is one of intrinsic properties of each topic. Some topics such as *big+ben* are stationary and coherent whereas others like *korean* are highly diverse and variant.

**Outbreak Detection of Subtopics.** The outbreak detection is important in Web mining since it reflects the change of information flows and people’s interests. We perform the outbreak detection by calculating an information-theoretic measure of link statistics. Note that the consecutive posterior sets are

**Table 1.** 47 topics of our Flickr dataset. The numbers in parentheses indicate the numbers of downloaded images per topic. 9,751,651 images are gathered in total.

Nation	<i>brazilian</i> (119,620), <i>jewish</i> (165,760), <i>korean</i> (254,386), <i>swedish</i> (94,390), <i>spanish</i> (322,085)
Place	<i>amazon</i> (160,008), <i>ballpark</i> (340,266), <i>big+ben</i> (131,545), <i>grandcanyon</i> (286,994), <i>pisa</i> (174,591), <i>wall+street</i> (177,181), <i>white+house</i> (241,353)
Animal	<i>butterfly+insect</i> (69,947), <i>cardinals</i> (177,884), <i>giraffe+zoo</i> (53,591), <i>jaguar</i> (122,615), <i>leopard</i> (121,061), <i>lobster</i> (144,596), <i>otter</i> (113,681), <i>parrot</i> (175,895), <i>penquin</i> (257,614), <i>rhino</i> (96,799), <i>shark</i> (345,606)
Object	<i>classic+car</i> (265,668), <i>keyboard</i> (118,911), <i>motorbike</i> (179,855), <i>pagoda</i> (128,019), <i>pedestrian</i> (112,116), <i>sunflower</i> (165,090), <i>television</i> (157,033)
Activity	<i>picnic</i> (652,539), <i>soccer</i> (1,153,969), <i>yacht</i> (225,508)
Abstract	<i>advertisement</i> (84,521), <i>economy</i> (61,593), <i>emotion</i> (119,899), <i>fine+art</i> (220,615), <i>horror</i> (157,977), <i>hurt</i> (141,249), <i>politics</i> (181,836)
Hot topic	<i>apple</i> (713,730), <i>earthquake</i> (65,375), <i>newspaper</i> (165,987), <i>simpson</i> (106,414), <i>starbucks</i> (169,728), <i>tornado</i> (117,161), <i>wireless</i> (139,390)



**Fig. 4.** Examples of subtopic evolution of *korean* and *big+ben* topics. Each column shows the clusters of each  $s_t$ . From top to bottom, we show top three out of five clusters of each  $s_t$  with average images (the first row) and top-four highest ranked images in the cluster (the second row). The *big+ben* is relatively stationary and coherent whereas the *korean* topic is highly dynamic and contains diverse subtopics such as *sports*, *food*, *buildings*, *events*, and *Korean War Memorial Park*.

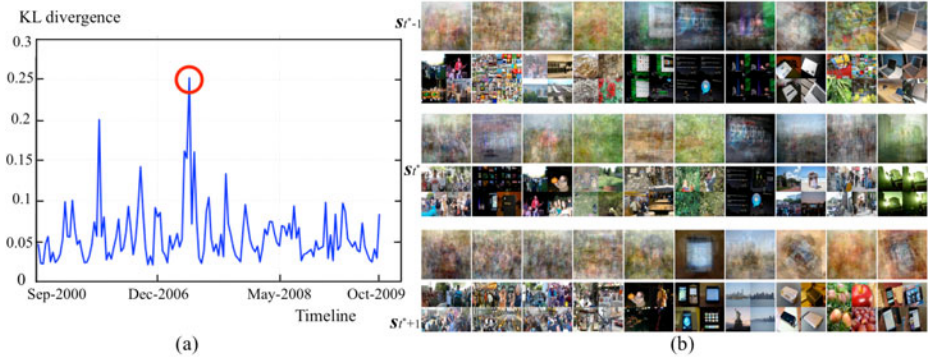
linked in our network. (*i.e.*  $s_{t-1}$  is connected to  $s_t$ , which is linked to  $s_{t+1}$ .) The basic idea of our outbreak detection is that if the subtopic distributions at step  $t-1$  and  $t+1$  are different each other, then the degree distribution of  $s_t$  to  $s_{t-1}$  ( $\mathbf{f}_{t,t-1}$ ) and the degree distribution of  $s_t$  to  $s_{t+1}$  ( $\mathbf{f}_{t,t+1}$ ) are dissimilar as well. For example, suppose that the dominant subtopic of  $s_{t-1}$  is *fruit apple* but the dominant one of  $s_{t+1}$  is *iphone*. Then, the degree of a *fruit apple* image  $i$  in  $s_t$  has high  $\mathbf{f}_{t,t-1}(i)$  but low  $\mathbf{f}_{t,t+1}(i)$ . On the other hand, an *iphone* image  $j$  in  $s_t$  has high  $\mathbf{f}_{t,t+1}(j)$  but low  $\mathbf{f}_{t,t-1}(j)$ . Both  $\mathbf{f}_{t,t-1}$  and  $\mathbf{f}_{t,t+1}$  are  $|s_t| \times 1$  histograms, each element of which is the sum of edge weights of a vertex in  $s_t$  with  $s_{t-1}$  and  $s_{t+1}$ , respectively. In order to measure the difference between  $\mathbf{f}_{t,t-1}$  and  $\mathbf{f}_{t,t+1}$ , we use *Kullback-Leibler* (KL) divergence in Eq 2.

$$D_{KL}(\mathbf{f}_{t,t+1} \parallel \mathbf{f}_{t,t-1}) = \sum_{i \in s_t} \mathbf{f}_{t,t+1}(i) \log \frac{\mathbf{f}_{t,t+1}(i)}{\mathbf{f}_{t,t-1}(i)} \quad (2)$$

Fig 5(a) shows an example of KL divergence changes along the 142 steps of *apple* tracking. The peaks of KL divergence indicate the radical subtopic changes from  $s_{t-1}$  to  $s_{t+1}$ . We observed the highest peak at the step  $t^* = 63$ , where  $s_{t^*}$  is distributed in [May-2007, Jun-2007]. Fig 5(b) represents ten subtopics of  $s_{t^*-1}$ ,  $s_{t^*}$ , and  $s_{t^*+1}$ , which are significantly different each other.

### 3.3 Comparison with Text Analysis

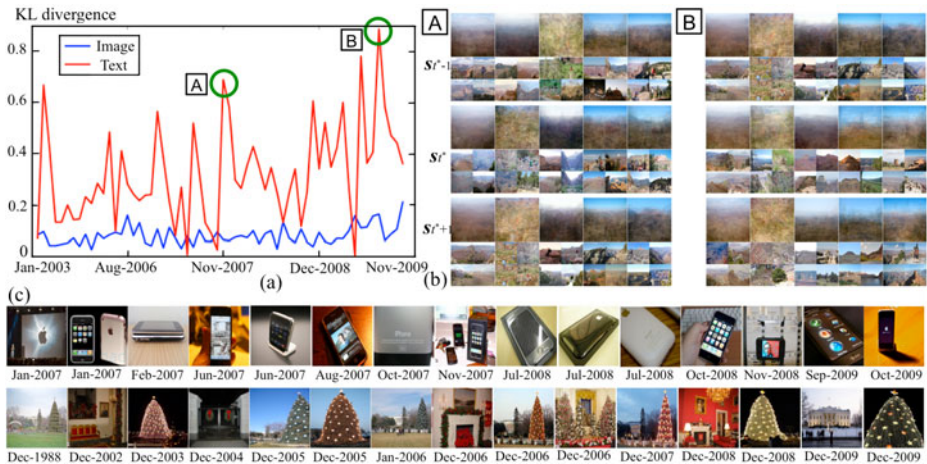
In this section, we empirically compare the image-based topic analysis with the text-based one. One may argue that the similar observations can be made from both images and the associated texts. However, our experiments show that the



**Fig. 5.** The outbreak detection of subtopics. (a) The variation of KL divergences for the *apple* topic. The highest peak is observed at the step  $t^*=63$  ([May-2007, Jun-2007] with the median of 11-Jun-2007). (b) The subtopic changes around the highest peak. Ten subtopics of  $s_{t^*-1}$ ,  $s_{t^*}$ , and  $s_{t^*+1}$  are shown from top to bottom. In each set, the first row shows average images of top 15 images and the bottom row shows top four highest ranked ones in each subtopic. In  $s_{t^*-1}$  and  $s_{t^*}$ , several subtopics about *Steve Jobs's presentation* are detected but disappear in  $s_{t^*+1}$ . Rather, *crowds in street* (*i.e.* 1st  $\sim$  4th clusters) and *iphone* (*i.e.* 6,8,10-th clusters) newly emerge in  $s_{t^*+1}$ .

associated texts do not overshadow the importance of information from the images. First of all, 13.70% of images in our dataset have no tags. It may be natural since the Flickr is oriented toward image sharing and thus text annotations are much less cared by users. In order to compare the dynamic behaviors detected from images and texts, we apply the outbreak detection method in previous section to both images and their associated tags. The only difference between them is the features: the spatial pyramids of SIFT and HOG for images and term frequency histograms for texts. Fig. 6 (a) shows an example of outbreak detection using images and texts for the *grandcanyon* topic, which is one of the most stationary and coherent topics in our dataset (*i.e.* no matter when the images are taken, the majority of them are taken for the scene of the *Grand Canyon*). The image-based analysis is able to successfully detect its intrinsic stationary behavior. However, the text tags are highly fluctuated mainly because tags are subjectively assigned by different users with little consensus. This is a well-known noise source of the images from the Web image search, and our result can be its another supporting example from the dynamics view.

Another important advantage of image-based temporal analysis is that it conveys more delicate information that is hardly captured by text descriptions. Fig. 6 (b) shows two typical examples about periodic updates of objects and events. For example, when a new *iphone* is released, the emergence of the *iphone* subtopic can be detected in the *apple* via both images and texts. However, the images can more intuitively reveal the upgraded appearance, new features, and visual context around the new event.



**Fig. 6.** The comparison between the topical analysis on the images and associated text tags. (a) The variation of KL divergences for the *grandcanyon* topic. The KL divergences of images are stationary along the timeline whereas those of texts are highly fluctuated. (b) The subtopic changes around the two highest peaks **A** (05-Nov-2007) and **B** (16-Aug-2009). Five subtopics of  $s_{t^*}^-$ ,  $s_{t^*}$ , and  $s_{t^*}^+$  are shown from top to bottom. Very little visual variation is observed between them. (c) 15 selected images tagged by *apple+new+iphone* (the first row) and *whitehouse+christmas* (the second row). They are sorted on the timeline.

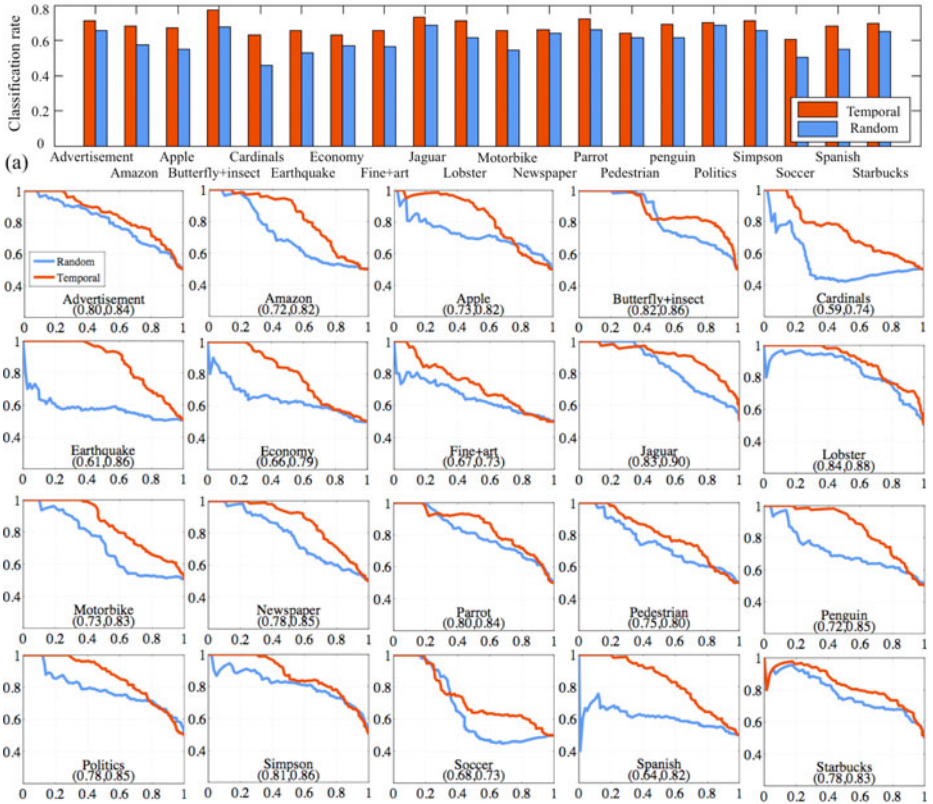
### 3.4 Temporal Association for Classification

As pointed in neuroscience research [19,21], human perception tends to strongly connect temporally smoothed visual information. Inspired by these studies, we perform preliminary tests to see whether it holds in Web images as well; The subtopics that consistently appear along the timeline can be more closely related to the main topic rather than the ones that are observed for only a short period. For example, the *fruit apple* is likely to consistently exist in the *apple* image set, which may be a more representative subtopic of the *apple* rather than a specific model of an early *Mac* computer. In this experiment, we generate two training sets from the extremely noisy Flickr images and compare their classification performance; The first training set is constructed by choosing the images that are temporally and visually associated, and the other set is generated by the random selection without temporal context.

Since our similarity network links temporally close and visually similar images, dominant subtopics correspond to large clusters and their central images map to hub nodes in the graph. The stationary probability is a popular ranking measure, and thus the images with high stationary probabilities can be thought of temporally and visually strengthened images. However, the proposed network representation is incomplete in the sense that images are connected in an only local temporal space. In order to cope with this underlying uncertainty, we generate training sets by the Metropolis-Hasting (MH) algorithm.

We first compute the stationary probability  $\pi_G$  of the network  $G$ . Since a general suggestion for a starting point in the MH is to begin around the modes of the distribution, we start from an image  $\theta_o$  that has the highest  $\pi_G(\theta)$ . From a current  $\theta$  vertex, we sample a next candidate point  $\theta^*$  from a proposal distribution  $q(\theta_1, \theta_2)$  that is based on a random surfer model as shown in Eq.3; the candidate is chosen by following an outgoing edge of the  $\theta$  with probability  $\lambda$ , but restarting it with probability  $1 - \lambda$  according to the  $\pi_G$ . A larger  $\lambda$  weights more the local link structure of the network while a smaller  $\lambda$  relies on  $\pi_G$  more. The new candidate is accepted with probability  $\alpha$  in Eq.3 where  $\tilde{w}_{ij}$  is the element  $(i, j)$  in the row-normalized adjacency matrix of  $G$ . We repeat this process until the desired numbers of training samples are selected.

$$\alpha = \min \left( \frac{\pi_G(\theta^*)q(\theta^*, \theta_{t-1})}{\pi_G(\theta_{t-1})q(\theta_{t-1}, \theta^*)}, 1 \right) \text{ where } q(i, j) = \lambda \tilde{w}_{ij} + (1 - \lambda)\pi_G(j) \quad (3)$$



**Fig. 7.** Comparison of the binary classification performance between *Temporal* training and *Random* training. (a) Classification accuracies of selected 20 topics. (b) Corresponding Precision-Recall curves. The number  $(n, m)$  underneath the topic name indicates the average precision of  $(Random, Temporal)$ .

We perform binary classification using the 128 nearest neighbor voting [20] in which we use the same descriptors and the cosine similarity in section 2.1. We generate the positive training set of each topic in two different ways; We sample 256 images by the MH method (called *Temporal* training) and randomly choose the same number of images (called *Random* training). For the negative training images, we randomly draw 256 images from the other topics of Flickr dataset. For the test sets, we downloaded 256 top-ranked images for each topic from Google Image Search by querying the same word in Table 1. The Google Image Search provides relatively clean images in the highest ranking. Since we would like to test whether the temporally associated samples are better generalization of the topic, the Google test sets are more suitable to our purpose than the images from the noisy Flickr dataset. In the binary classification test of each topic, the positive test images are the 256 Google images of the topic and the negative test images are 256 Google images that are randomly selected from the other topics. Note that in each run of experiment, only the positive training samples are different between *Temporal* and *Random* tests. The experiments are repeated ten times, and the mean scores are reported.

Fig. 7 summarizes the comparison of recognition performance between *Temporal* and *Random* training. Fig. 7(a) shows the classification rates for the selected 20 topics. The accuracies of *Temporal* training are higher by 8.05% on average. Fig. 7(b) presents the corresponding precision-recall curves, which show that the *temporal association* significantly improves the confidence of classification. The *Temporal* training is usually better than the *Random* training in performance, but the improvement is limited in some topics; In highly variant topics (e.g. *advertisement* and *starbucks*), the temporal consistency is not easily captured. In stationary and coherent topics (e.g. *butterfly+insect* and *parrot*), the random sampling is also acceptable.

## 4 Discussion

We presented a nonparametric modeling and analysis approach to understand the dynamic behaviors of Web image collections. A sequential Monte Carlo based tracker is proposed to capture the subtopic evolution in the form of the similarity network of the image set. In order to show the usefulness of the image-based temporal topic modeling, we examined subtopic evolution tracking, subtopic outbreak detection, the comparison with the analysis on the associated texts, and the use of temporal association for recognition improvement. We believe that this line of research has not yet fully explored and various challenging problems still remain unsolved. In particular, more study on the temporal context for recognition may be promising.

**Acknowledgement.** This research is supported in part by funding from NSF IIS-0713379, DBI-0546594, Career Award, ONR N000140910758, DARPA NBCH 1080007 and Alfred P. Sloan Foundation awarded to Eric P. Xing, and NSF Career Award IIS 0747120 to Antonio Torralba.

## References

1. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking. *IEEE Trans. Signal Processing* 50(2), 174–188 (2002)
2. Becker, S.: Implicit Learning in 3D Object Recognition: The Importance of Temporal Context. *Neural Computation* 11(2), 347–374 (1999)
3. Blei, D.M., Lafferty, J.D.: Dynamic Topic Models. In: *ICML* (2006)
4. Bosch, A., Zisserman, A., Munoz, X.: Image Classification using Random Forests and Ferns. In: *ICCV* (2007)
5. Boutell, M., Luo, J., Brown, C.: A Generalized Temporal Context Model for Classifying Image Collections. *Multimedia Systems* 11(1), 82–92 (2005)
6. Cao, L., Luo, J., Kautz, H., Huang, T.S.: Annotating Collections of Photos using Hierarchical Event and Scene Models. In: *CVPR* (2008)
7. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge, VOC 2010 Results (2010), <http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html>
8. Hinton, G.E.: Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation* 14(8), 1771–1800 (2002)
9. Isard, M., Blake, A.: CONDENSATION – Conditional Density Propagation for Visual Tracking. *Int. J. Computer Vision* 29(1), 5–28 (1998)
10. Kalogerakis, E., Vesselova, O., Hays, J., Efros, A., Hertzmann, A.: Image Sequence Geolocation with Human Travel Priors. In: *ICCV* (2009)
11. Kim, G., Torralba, A.: Unsupervised Detection of Regions of Interest using Iterative Link Analysis. In: *NIPS* (2009)
12. Li, Y., Crandall, D.J., Huttenlocher, D.P.: Landmark Classification in Large-scale Image Collections. In: *ICCV* (2009)
13. Liu, C., Yuen, J., Torralba, A.: Nonparametric Scene Parsing: Label Transfer via Dense Scene Alignment. In: *CVPR* (2009)
14. Liu, C., Yuen, J., Torralba, A., Sivic, J., Freeman, W.T.: SIFT Flow: Dense Correspondence across Different Scenes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part III*. LNCS, vol. 5304, pp. 28–42. Springer, Heidelberg (2008)
15. MacKay, D.: *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Cambridge (2002)
16. Paletta, L., Prantl, M., Pinz, A.: Learning Temporal Context in Active Object Recognition Using Bayesian Analysis. In: *ICPR* (2000)
17. Quack, T., Leibe, B., Gool, L.V.: World-scale Mining of Objects and Events from Community Photo Collections. In: *CIVR* (2008)
18. Russell, B.C., Torralba, A.: Building a Database of 3D Scenes from User Annotations. In: *CVPR* (2009)
19. Sinha, P., Balas, B., Ostrovsky, Y., Russell, R.: Face Recognition by Humans: Nineteen Results All Computer Vision Researchers Should Know About. *Proceedings of the IEEE* 94(11), 1948–1962 (2006)
20. Torralba, A., Fergus, R., Freeman, W.T.: 80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition. *IEEE PAMI* 30(11), 1958–1970 (2008)
21. Wallis, G., Bulthöff, H.H.: Effects of Temporal Association on Recognition Memory. *PNAS* 98(8), 4800–4804 (2001)
22. Wang, X., McCallum, A.: Topics Over Time: a Non-Markov Continuous-Time Model of Topical Trends. In: *KDD* (2006)

# Non-local Characterization of Scenery Images: Statistics, 3D Reasoning, and a Generative Model

Tamar Avraham and Michael Lindenbaum

Computer science department, Technion - I.I.T., Haifa 3200, Israel  
{tammya,mic}@cs.technion.ac.il

**Abstract.** This work focuses on characterizing scenery images. We semantically divide the objects in natural landscape scenes into background and foreground and show that the shapes of the regions associated with these two types are statistically different. We then focus on the background regions. We study statistical properties such as size and shape, location and relative location, the characteristics of the boundary curves and the correlation of the properties to the region’s semantic identity. Then we discuss the imaging process of a simplified 3D scene model and show how it explains the empirical observations. We further show that the observed properties suffice to characterize the gist of scenery images, propose a generative parametric graphical model, and use it to learn and generate semantic sketches of new images, which indeed look like those associated with natural scenery.

## 1 Introduction

*By age 5 or 6 children develop a set of symbols to create a landscape that eventually becomes a single variation repeated endlessly. A blue line and sun at the top of the page and a green line at the bottom become symbolic representations of the sky and ground.* From: Drawing on the Right Side of the Brain. Betty Edwards, 1979 [1].

When we think of “scenery” or “natural landscape” images, we typically imagine a photograph or a painting, with a few horizontal background regions, each spanning the frame. The highest region would usually be the sky, while the lower regions might include mountains, trees, flowers, water (lake/sea), sand, or rocks. This work examines whether this intuition is justified, by analyzing image statistics and by modeling the 3D world and analyzing its 2D projections as imaged in typical scenery photography. We semantically divide the objects in natural landscape scenes into background and foreground and show that the shapes of the regions associated with these two types are statistically different. We then focus on the background regions. We study statistical properties such as size and shape, location and relative location, the characteristics of the boundary curves and the correlation of the properties to the region’s semantic identity.



These properties, which could be characterized as common world knowledge, have been used, in part, to enhance several computer vision algorithms. Nonetheless, they have not, to the best of our knowledge, been explicitly expressed, summarized, or computed.

This paper makes three contributions: First, we make several observations about image region properties, and collect empirical evidence supporting these observations from annotated segmentations of 2D scenery images (Section 2). Second, we discuss the imaging process of a simplified 3D scene model and show how it explains the empirical observations (Section 3). In particular, we use slope statistics inferred from topographic maps to show why land regions whose contour tangents in aerial images are statistically uniformly distributed appear with a strong horizontal bias in images taken from ground level. Third, we show that the observed properties suffice to characterize the gist of scenery images: In Section 4 we propose a generative parametric graphical model, and use it to learn and generate semantic sketches of new images, which indeed look like those associated with natural scenery. The novel characteristics analyzed in this work may improve many computer vision applications. In Section 5 we discuss our future intentions to utilize them for the improvement of top-down segmentation, region annotation, and scene categorization.

## 1.1 Related Work

Statistics of natural images play a major role in image processing and computer vision; they are used in setting up priors for automatic image enhancement and image analysis. Previous studies in image statistics (e.g., [2,3,4,5]) mostly characterized local low-level features. Such statistics are easy to collect as no knowledge about high-level semantics is required. Lately, computer-vision groups have put effort into collecting human annotations (e.g., [6,7,8]), mostly in order to obtain large ground-truth datasets that enable the enhancement and validation of computer vision algorithms. The availability of such annotations enables the inference of statistics on semantic characteristics of images. A first step was presented in [6] where statistics on characteristics of human segmentation were collected. In [7] a few interesting statistics were presented, but they mainly characterize the way humans segment and annotate. We follow this direction relying on human annotations to suggest characteristics that quantify high-level semantics.

The importance of context in scene analysis was demonstrated a while ago [9] and used intensively in recent years for improving object detection by means of cues pertaining to spatial relations and co-occurrence of objects [10,11,12], annotated segments [13], or low-level scene characteristics and objects [14,15]. Part of the work presented here focuses on the co-occurrence of and spatial relations between background objects. Objects are semantically divided into background and foreground, implying that image analysis applications may benefit from treating objects of these two classes differently. This is related to the *stuff* vs. *things* concept [16].

We found that the background-region boundary characteristics correlate with the identity of the lower region. This observation is consistent with the

observation made in figure-ground assignment studies, that of the two regions meeting in a curve, the lower region is more likely to be the "figure", i.e., of lower depth [17,18]. Our work is also related to the recent successful approach which uses learning to model the relation between image properties and the corresponding 3D structure (e.g., [19,20,21]). The approach in [21], for example, associates the images with particular classes of geometrically specified 3D structures. We focus here on the wide class of scenery images with a variety of semantically specified regions, and provide an image model (supported by 3D scene analysis) characterizing the large scale image properties.

## 2 Observations and Evidence

In this section we present some observations on the appearance of background regions in landscape images. After showing how the statistics of their general shape differ from those of foreground objects, we discuss their relative location and characteristics of their contours.

We use fully annotated landscape images included in the Labelme toolbox [7]. Of the images used in [22], where outdoor images were divided into eight categories, we used all the images of the three natural landscape categories: *coast*, *mountain*, and *open country* (for a total of 1144 256X256 images).

With the Labelme toolbox, a Web user marks polygons in the image and freely provides a textual annotation for each. This freedom encourages the use of synonyms and spelling mistakes. Following [7], synonyms were grouped together and spelling mistakes were corrected. (For details see [23].)

### 2.1 The General Shape of Background vs. Foreground Objects

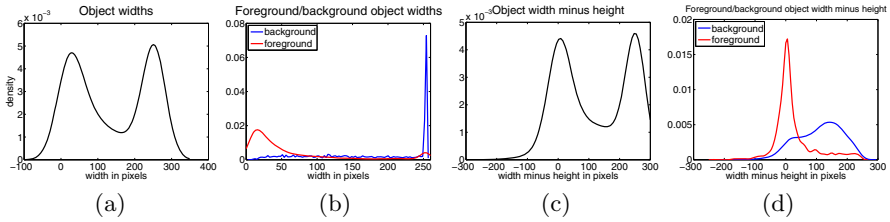
We semantically divide the annotated objects into two sets: *background objects* and *foreground objects*. The background set includes all objects belonging to the following list: *sky, mountain, sea, trees, field, river, sand, ground, grass, land, rocks, plants, snow, plain, valley, bank, fog bank, desert, lake, beach, cliff, floor*. Foreground objects are defined as those whose annotation does not belong to that list. (Note that while *trees, rocks, and plants* are considered *background objects*, *tree, rock, and plant* are considered *foreground objects*.) For a summary of the occurrence of each of the background and foreground labels see [23]. Differences in the distribution of the size and aspect ratios of the bounding box of these two classes give rise to the following observations:

**Observation 1: Many background objects exceed the image width.**

The background objects are often only partially captured in the image. See Fig. 1(a) and Fig. 1(b) for background vs. foreground object width statistics. Note the sharp bimodality of the distribution.

**Observation 2: The background objects are wide and of low height while foreground objects' shape tend to be isotropic.**

Although the entire background object width is usually not captured, the height of its annotated polygon is usually small relative to the height of the image. See



**Fig. 1.** Bounding boxes of imaged background objects are usually low height horizontal rectangles, while those of imaged foreground objects tend to be squares: (a) width density estimation (by kernel smoothing) of annotated objects from the Labelme dataset (where width is the difference in pixel units between the rightmost and the leftmost points in the annotated polygon. All images are  $256 \times 256$ ); (b) width density estimation of background and foreground objects taken separately; (c) width minus height density estimation of annotated objects; (d) width minus height density estimation of background and foreground objects taken separately. The distributions in (a),(c) were generated by an equal number of foreground and background objects. A random subset of background objects was used to compensate for the larger number of background objects in this dataset.

Fig. 1(c) and Fig. 1(d): the width and height difference of foreground objects is distributed normally with zero mean, while the width and height difference of background objects significantly favors width. This implies that bounding boxes of imaged background objects are low height horizontal rectangles, while bounding boxes of imaged foreground objects tend to be squares. (Note that all the images in this dataset are squares, so the horizontal bias is not due to the image dimensions.) See further analysis and discussion on the horizontalness of background regions in Section 3.

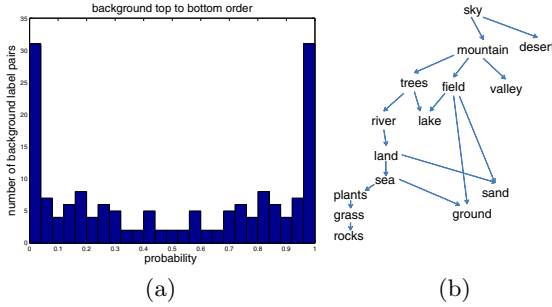
## 2.2 The Top-Down Order of Background Objects

Because background objects tend to be wide—frequently spanning the image horizontally though not vertically—each landscape image usually includes a few background regions, most often appearing one on top of the other.

**Observation 3: The relative locations of types of background are often highly predictable.**

It is often easy to guess which background type will appear above another. For instance, if an (unseen) image includes sky and ground, we know that the sky will appear above the ground. Here we extend this ostensibly trivial “sky is above” observation and test above-and-below relations for various pairs of background types. Let  $\mathcal{I}$  denote the set of all landscape images. Let  $A - B$  denote that background type  $A$  appears above background type  $B$  in image  $I \in \mathcal{I}$  (e.g.,  $A$ =trees,  $B$  = mountain). We estimate the probability for  $A$  to appear above  $B$  (or  $B$  to appear above  $A$ ), given that we know both appear in an image:

$$p_{A-B} = p(A - B | A, B \in I) \simeq \frac{|\{I \in \mathcal{I} | A, B \in I, A - B\}|}{|\{I \in \mathcal{I} | A, B \in I\}|}. \quad (1)$$



**Fig. 2.** Expected relative location of background regions. For most background type pairs, there is a strong preference for one type to appear above the other. (a) The probability for a background region of identity  $A$  to appear above a background region with identity  $B$ , summarized in a histogram for various background identity pairs. (b) Topological ordering of background identities can be defined: this DAG (Directed Acyclic Graph) is associated with the reachability relation  $R : \{(A, B) | p_{A-B} > 0.7\}$ .

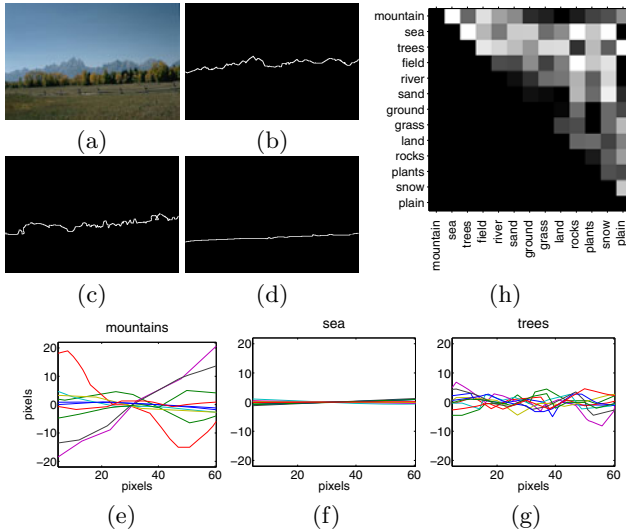
See Fig. 2 for a histogram of  $p_{A-B}$  for  $A$  and  $B$  being two background identities,  $A \neq B$ . The histogram is symmetric as  $p_{A-B} + p_{B-A} = 1$ . There are 22 background categories. Out of 231 possible pairs, only 116 appear at least once in the same image. The histograms consider only pairs that coappeared at least 5 times (83 pairs). Most pairs show a clear preference for the more likely relative location. The most obvious is sky, which appears above all other background categories. However, some examples for pairs for which  $p_{A-B} > 0.9$  are mountain-lake, trees-plants, mountain-beach, trees-rocks, plain-trees. For 84% of the pairs,  $\max(p_{A-B}, p_{B-A}) > 0.7$ . The dominant order relations induce a (partial) topological ordering of the background identities and can be described by a DAG (Directed Acyclic Graph). The DAG in Fig. 2(b) is associated with the reachability relation  $R : \{(A, B) | p_{A-B} > 0.7\}$ , i.e., there is a directed path in the graph from  $A$  to  $B$  if and only if  $A$  appears above  $B$  in more than 70% of the images in which they coappear. As evident here, learning the typical relative locations of background regions is informative.

### 2.3 Contours Separating Background Regions

If a background region  $A$  appears above a background region  $B$ , it usually means that  $B$  is closer to the photographer and is partly occluding  $A$  [17, 18]. Hence, we can say that a contour separating background regions is usually the projection of the closer (lower) background region’s silhouette, and usually has characteristics that can be associated with this background type.

**Observation 4: The characteristics of a contour separating two background regions correlates with the lower region’s identity.**

Consider Fig. 3. The curves in Fig. 3(b-d) are associated with the background object classes ‘mountain’, ‘trees’, and ‘grass’, respectively. See also Fig. 3(e)-(g), for contours associated with different background objects. When the lower



**Fig. 3.** Characteristics of background region boundaries. (a)-(d) An image and its hand segmentation [6]. (e)-(g) A few sample contour segments associated with background object classes ‘mountain’, ‘sea’, and ‘trees’ (from Labelme). (h) Classification accuracies for two-class background identity, based only on the appearance of the region’s upper boundary. The accuracy of classification is displayed in linear gray scale (black for all values below 0.5 and white for all values above 0.8).

background object is of type sea, grass or field, the boundary is usually smooth and horizontal, resembling a DC signal. For background objects such as trees and plants, the boundary can be considered as a high frequency 1D signal. For background objects of type ‘mountain’, the associated boundaries usually resemble 1D signals of rather low frequency and high amplitude.

Adopting a signal representation, we checked how informative the contours are for discriminating between background identities: The Labelme landscape images were randomly divided into equally sized training and validation sets. For each background labeled region, the upper part of its contour was extracted and cut to chunks of 64-pixels length. Each chunk was FFT transformed, and the norms of 32 coefficients (2-33) were added to a training or validation set associated with the background label. Only labels for which the training set included at least 30 ‘signals’ were further considered. For each pair of labels we checked whether the two associated classes could be distinguished using ONLY the upper part of their boundary. We used an SVM classifier with an RBF kernel. (To avoid bias due to different class size, which would have made discrimination easier, we took equal sized training sets and validation sets from the two classes, by randomly selecting some members of the larger set.) Fig. 3(h) summarizes the accuracies of the two-class classifiers. While the information in the contour’s shape cue discriminates well between several pairs of background types, it cannot discriminate between all pairs. Better results may be obtained by adding local

properties as discussed in Section 5. In Section 4 we show that the contour shape information together with the relative location may be used to specify a generative model that captures the gist of scenery images.

### 3 Why Are Background Regions Horizontal? A 3D Analysis

In Section 2 we have statistically shown that imaged land region boundaries have a strong horizontal bias. To account for this empirical finding, we model the 3D world and analyze its 2D projections imaged in typical scenery photography. We start by a simplified ‘flatland’ model, continue by considering also land coverage (e.g., vegetation), and finally terrain elevation. In all these cases, we show why land regions whose contour tangents in aerial images are uniformly distributed appear with a strong horizontal bias in scenery filmed on the ground.

#### 3.1 Flatland

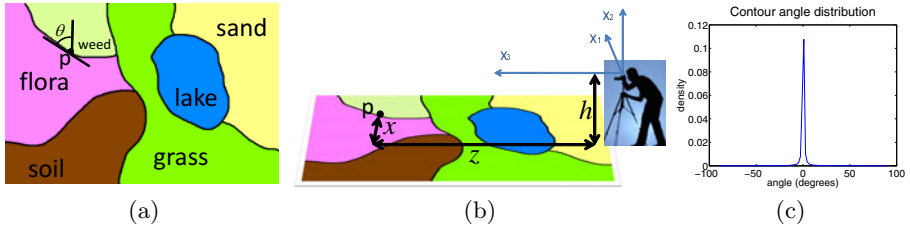
*Place a penny on the middle of one of your tables in Space ... look down upon it. It will appear a circle....gradually lower your eyes ... and you will find the penny becoming more and more oval to your view....* From Flatland, by Edwin A. Abbott, 1884 [24].

We first consider a natural terrain in a flat world with no mountains, no valleys, and vegetation of zero height. This terrain may be divided into a few regions, each with different “clothing”, as depicted from an aerial view in Fig. 4(a). Consider the contour lines dividing the different regions. Let  $\Theta$  be the set of tangent angles for all such contours, measured relative to some arbitrary 2D axis on the surface. It is reasonable to assume that the angles in  $\Theta$  are uniformly distributed in the range  $[0^\circ, 360^\circ)$ . Now consider a person standing on that surface at an arbitrary point, taking a picture. Let  $\Theta'$  be the set of angles that are the projections of the angles in  $\Theta$  on the camera’s image plane. How is  $\Theta'$  distributed?

For simplicity, we adopt the pinhole camera model. Let a point  $p$  on a contour line be located at  $(x, -h, z)$  in a 3D orthogonal coordinate system originating at the camera pinhole. (See Fig. 4(b).) Redefine  $\theta$  as the angle of the tangent to the contour line at  $p$ , relative to the 1st axis. The angle  $\theta'$ , associated with the projected contour on the camera’s sensor is

$$\tan \theta' = \frac{h \tan \theta}{z - x \tan \theta} . \quad (2)$$

For details see [23]. See Fig. 4(c) for a plot of the distribution of  $\Theta'$ . The strong peak around  $0^\circ$  explains why background regions tend to be wide and horizontal in scenery images (as statistically shown in Section 2.1).



**Fig. 4.** The tangents of background object contours when imaging a flat terrain. (a) A schematic illustration of an aerial image; (b) a view from height  $h$ . A point  $p$  that lies on a land region boundary is located at  $(x, -h, z)$  relative to a 3D orthogonal coordinate system originating at the camera pinhole; (c) the distribution of the tangent of boundary lines in such an image, assuming that the tangents of aerial image boundaries are uniformly distributed,  $\theta \sim U[0, 180]$ ,  $h = 2[\text{m}]$ ,  $z \sim U[0[\text{m}], 1000[\text{m}]]$ , and  $x \sim U[0[\text{m}], 500[\text{m}]]$ .

### 3.2 Land Cover

Now we extend the flatland model and consider a flat terrain with protruding coverage, e.g., sand, gravel, or rock covered regions, fields of flowers, or even forests. Each such region's cover is often of approximately equal height. Then, the profile of this land (slicing through any vertical plane) can be considered as a piecewise constant function.

Consider again the photographer at an arbitrary point on the flat terrain. First consider the case where the cover is lower than the camera (e.g., bushes, pebbles). The cover of a raised region would occlude part of the more distant regions. The distribution of angles associated with the tangents of imaged contours describing the upper contour of such cover is even more concentrated near the origin, as the height difference of points on the cover and the pinhole is smaller compared to the height difference in flatland. When the land cover is higher than the camera (e.g., forest), the region cannot be viewed from above and only the side facing the camera will be captured. Angles on the upper contour, at height  $H$  project to image angles,  $\theta'$ , where  $\tan \theta' = \frac{(H-h) \tan \theta}{z - x \tan \theta}$ . Typically, trees are only a few meters high while the viewing distance  $z$  for landscape images is usually much larger. Therefore, the statistical shortening of the contour angles still holds.

Naturally, the land cover height is not constant, but characterized by some nominal value with small local perturbation. These perturbations may significantly change the local projected angle but not the general direction of the contour, which stays close to horizontal.

### 3.3 The World Is Wrinkled: Ground Elevation and Slope Statistics

Obviously, the earth's terrain is not flat. Its surface is wrinkled by mountains, hills and valleys. To approximately express how ground elevations affect the appearance of background object contours in images, we rely on a slope statistics database [25]. This dataset includes histograms over 8 variable size bins for each

land region of about 9 square kilometers. (The bins are nonuniform and are specified between the slope limits of 0%, 0.5%, 2%, 5%, 10%, 15%, 30%, 45%, and infinity.) We use the average histogram. To get a distribution, we approximate the slope distribution within the bin as uniform. See Fig. 5(b). We also use a histogram of the maximum slope over the land regions (Fig. 5(c)).

The slope statistics affect two landscape image contour types: (1) The contours of mountains associated with occluding boundaries (e.g., skylines). (2) The contours between different types of regions on the terrain.

The distribution depicted in Fig. 5(c) provides a loose upper bound for the expected distribution of projected tangent angles associated with the former set. Even so, the horizontal bias is apparent.

To account for the effect of ground elevation on the latter type of background contours, we extend the analysis suggested in Section 3.1. Instead of considering an angle  $\theta$  lying on a flat terrain, we consider  $\theta$  to lie on an elevated plane with slope gradient angle  $\varphi$ . See Fig. 5(a). The plane is rotated relative to the image plane, forming an angle  $\omega$  with the  $X_1$  axis. The point  $p$  is at height  $H$  relative to the camera height. The projected tangent angle  $\theta'$  is given by

$$\tan \theta' = \frac{H(\cos \theta \sin \omega + \sin \theta \cos \varphi \cos \omega) - z \sin \theta \sin \varphi}{x(\cos \theta \sin \omega + \sin \theta \cos \varphi \cos \omega) - z(\cos \theta \cos \omega - \sin \theta \cos \varphi \sin \omega)}. \quad (3)$$

For details see [23]. To get an idea how  $\theta'$  is distributed, we make several reasonable assumptions:  $\theta$  is uniformly distributed as before,  $\varphi$  is distributed as the slope angle distribution (Fig. 5(b)), and  $\omega$  is uniformly distributed  $U[-90^\circ, 90^\circ]$ . The distribution of  $H$  was estimated by sampling an elevation map [25], using the height difference between pairs of locations up to 9km apart. See an analytic plot of the distribution of  $\theta'$  in Fig. 5(d).

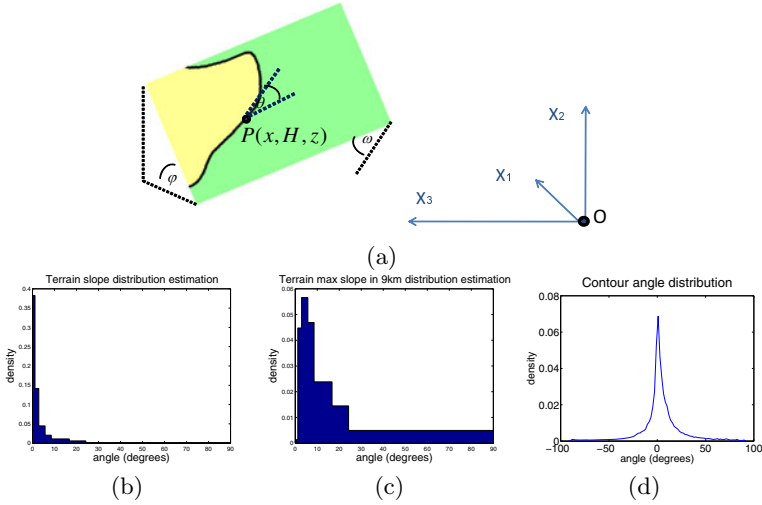
The above analysis isn't perfect from either a geometrical, a topographical, or an ecological point of view; e.g., we do not account for the roundness of the world, we assume that the camera is levelled with the ground, we assume independency between the slope steepness and the imaging height difference, and we do not consider dependencies between the steepness of slopes and the location of different land regions. For instance, the slope of a lake is always zero. Nevertheless, we believe the horizontal bias of background contours, as observed empirically, is sufficiently accounted for by the simplified analysis described here.

## 4 A Generative Model

The observations and the statistical quantification presented in Section 2 enable us to propose the following generative model for scenery image sketches. See, e.g., Fig. 3(a)-(d). Our model considers the top-down order of background regions, the relative area covered by each, and the characteristics of their boundaries, and assigns a probability for each possible annotation sequence.

Let  $S = (h_1, \dots, h_n, S_1, S_2, \dots, S_{n-1})$  be the description of a background segmentation for an image divided into  $n$  background segments, ordered from the highest in the image ( $i = 1$ ) to the lowest ( $i = n$ ).  $h_i$  is the mean height of region





**Fig. 5.** Distribution of background object contour angles in a “wrinkled” world. (a) A point  $p$  lies on a boundary between land regions. It is located on an elevated slope with gradient angle  $\varphi$ . The infinitesimal plane is rotated at an angle  $\omega$  relative to  $X_1$  axis. (b) Estimated terrain slope distribution using the IASA-LUC dataset [25]. (c) Estimated distribution of the maximum slope over land regions, each covering approximately 9 square kilometers. (d) The distribution of the tangents of imaged boundaries, following the analysis in the text.

$i, \sum_i h_i = 0$ .  $S_i$  describes a ‘1-D signal’ associated with the boundary between  $i$  and  $i + 1$ . Let  $\mathbf{l} = (l_1, \dots, l_n)$  be a labeling sequence, where  $l_i \in L$ , and  $L$  is the set of background labels. We shall use the approximated distribution

$$P(\mathbf{l}|S) = \frac{P(S|\mathbf{l})P(\mathbf{l})}{\sum_{\mathbf{l} \in L^n} P(S|\mathbf{l})P(\mathbf{l})} \propto P_1(\mathbf{l}) \prod_{i=1, \dots, n} P_2(h_i|l_i) \prod_{i=1, \dots, n-1} P_3(S_i|l_{i+1}) . \tag{4}$$

This approximation assumes that the height of a region depends only on its identity, and that a boundary’s characteristics depend only on the identity of the corresponding lower region. Other dependencies are ignored.

The next three sections discuss the distributions  $P_1, P_2$  and  $P_3$ .

### 4.1 A Markov Network for Modeling the Top-Down Label Order

$P_1$  is the probability of a scenery image annotation. We use a Markov network to represent possible label sequences. Let  $m = |L|$  be the number of possible background labels. The network has  $m + 2$  nodes (statuses). The first  $m$  are associated with the members of  $L$ . In addition, there is a starting status denoted ‘top’ and a sink status denoted ‘bottom’. Let  $M$  be the transition matrix.  $M(l_i, l_j)$  is the probability to move from status associated with label  $l_i$  to status associated with  $l_j$ , i.e., that a region labeled  $l_i$  appears above a region labeled

$l_j$  in an image.  $M(\text{top}, l_i)$  and  $M(l_i, \text{bottom})$  are the probabilities that a region with label  $l_i$  is at the top/bottom of an image, respectively. Then:

$$P_1(\mathbf{l} = (l_1, \dots, l_n)) = M(\text{top}, l_1) \prod_{i=1, \dots, n-1} M(l_i, l_{i+1}) M(l_n, \text{bottom}), \quad (5)$$

i.e., the probability for a labeling sequence  $(l_1, \dots, l_n)$  is equal to the probability for a random walk starting at the initial network state to go through the states corresponding with  $(l_1, \dots, l_n)$ , in that order, and to then continue to the sink status. We use a dataset of images for which the sequences of background labeling are known, e.g., the Labelme landscape images, and set the parameters of the model (i.e., the matrix  $M$ ) by counting the occurrences of the different ‘moves’.

## 4.2 A Normal Distribution for the Height Covered by Each Region

$P_2$  models the distribution of the relative height of an image region associated with a certain label. Here we simply use a normal distribution, learning the mean and variance of each region type’s relative height.

## 4.3 Modeling Background Contours with PCA

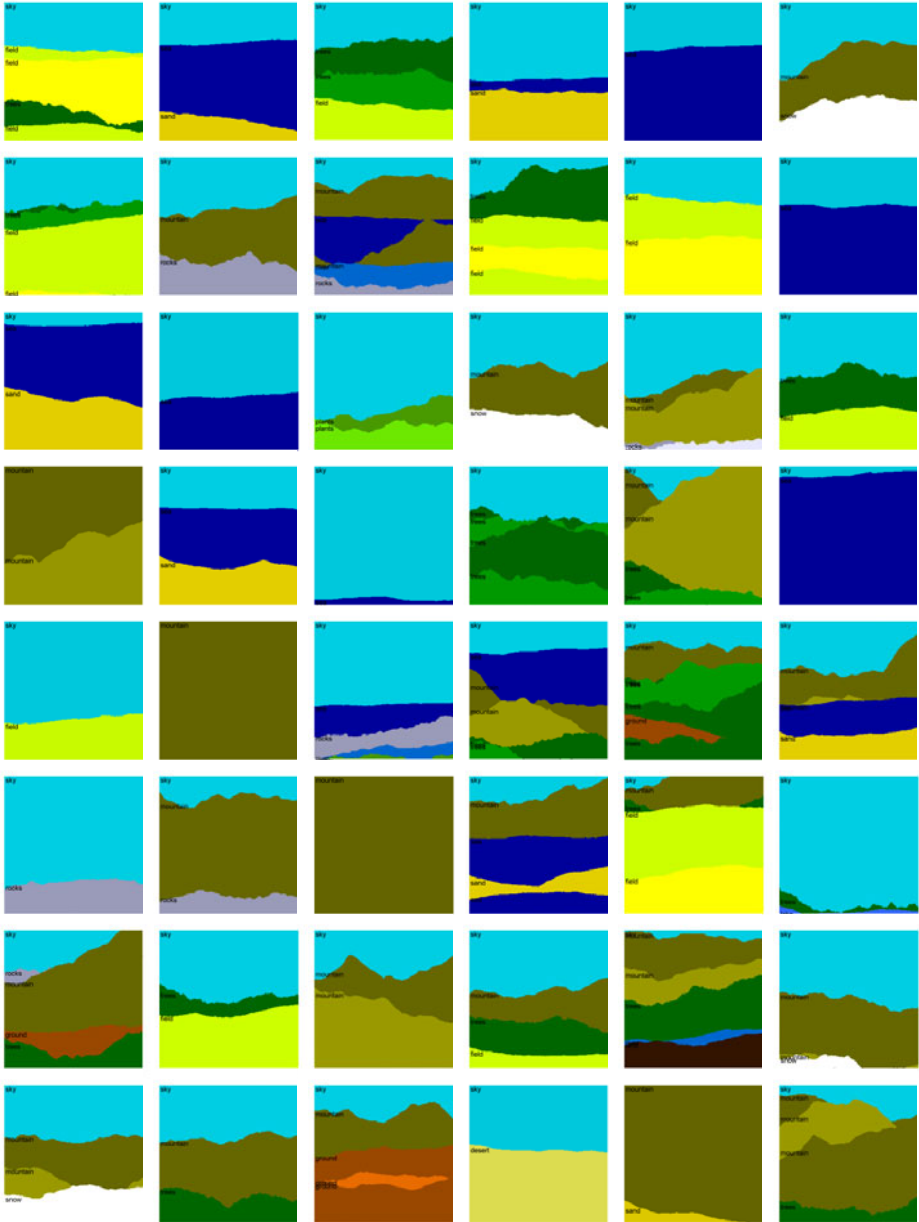
$P_3(S_i|l_k)$  is the probability of a contour with appearance  $S_i$  to separate two background regions, the lower being of type  $l_k$ . In Section 2.2 we have shown that  $S_i$  and  $l_k$  are correlated (observation 4). To estimate the probability from examples we use PCA approximation (Principle Component Analysis [26]). Given a training set of separation lines associated with background type  $l_k$ , each separation line is cut to chunks of 64-pixel length<sup>1</sup>. Each chunk’s mean value is subtracted, and PCA is performed, resulting in the mean vector  $\bar{\mu}$ , the first  $\kappa$  principle components  $\Phi$  (a  $64 \times \kappa$  matrix) and the corresponding eigen values  $\bar{\lambda} = (\lambda_1, \dots, \lambda_\kappa)$ .  $\kappa$  is chosen so that 95% of the variation in the training set is modeled.

The PCA modeling allows both computation of the probability of a new separating line  $S_i$ , cut to chunks  $S_{i1}, \dots, S_{im}$ , to belong to the learned distribution  $\Omega$ , and generation of new separating lines belonging to the estimated distribution.

## 4.4 Generative Model Demonstration

We can now use this model to generate sketches of new images. To generate a sketch, a sequence of labels is first drawn from a random walk on the transition matrix ( $P_1$ ). Then, heights are randomly picked from the normal distributions ( $P_2$ ) and normalized. Finally, for each separating line indexed  $i$ , four 64-length chunks  $S_{i,1}, \dots, S_{i,4}$  are generated by  $G = \bar{\mu} + \bar{b} \cdot \Phi$ , where  $b_j \sim N(0, \sqrt{\lambda_j})$ ,  $j = 1, \dots, \kappa$ . (Each chunk is generated independently, ignoring appearance dependencies between chunks of the same line.) The leftmost point of chunk  $S_{i,1}$

<sup>1</sup> Cutting the ‘signals’ into chunks also allows us to use separating lines from the training set that do not horizontally span the entire image or that are partly occluded by foreground objects. Moreover, it enlarges the training set, by obtaining a few training items (up to 4 chunks) from each separating contour.



**Fig. 6.** A random sample of ‘annotated’ landscape images generated by our model. The regions are colored with colors associated with their annotation (sky regions are colored in blue, ground regions are colored in brown, etc.) Best viewed on a color computer screen.

is placed at image coordinate  $(0, \sum_{j=1}^i h_j)$  (where  $(0, 0)$  is the top-left image corner). Chunk’s  $S_{i,m}$  ( $m = 2, 3, 4$ ) leftmost point is connected to the rightmost part of  $S_{i,m-1}$ . See Fig. 6 for a random sample of ‘annotated’ scenery landscape image sketches generated by the model.

To evaluate the generated, annotated images, we took a random sample of 50 and asked two participants naive to this research, aged 7 and 37, to say whether they seem to be the annotations of real landscape photos. The first participant answered ‘yes’ for 37 images, ‘no’ for 5, and was not sure about 8. The second participant answered ‘yes’ for 44 images, ‘no’ for 3 and ‘not sure’ for 3.

## 5 Discussion

This work focused on characterizing scenery images. Intuitive observations regarding the statistics of co-occurrence, relative location, and shape of background regions were explicitly quantified and modeled, and 3D reasoning for the bias to horizontalness was provided.

Our focus was on non-local properties. The generated image sketches, which seem to represent a wide variety of realistic images, suggest that the gist of such images is described by those properties. The proposed model provides a prior on scenery image annotation. In future work we intend to integrate local descriptors (see e.g. [27,28]) into our model and to then apply automatic annotation of segmented images. The large scale model introduced here should complement the local information and lead to better annotation and scene categorization [27]. Relating the contour characteristics to object identity can be useful for top-down segmentation (e.g., [13]). Specifically, it may address the “shrinking bias” of graph-cut-based methods [29].

A more complete model of scenery images may augment the proposed background model with foreground objects. Such objects may be modeled by location, size, shape, and their dependency in the corresponding properties of other co-occurring foreground objects and of the corresponding background regions.

One immediate application would be to use the probabilistic model to automatically align scenery pictures, similar to the existing tools for automatic alignment of scanned text. Some would find an artistic interest in the generated scenery sketches themselves, or may use them as a first step to rendering.

## References

1. Edwards, B.: Drawing on the Right Side of the Brain. Tarcher Publishing (1979)
2. Lee, A.B., Mumford, D., Huang, J.: Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model. IJCV 41, 35–59 (2001)
3. Srivastava, A., Lee, A., Simoncelli, E.: On advances in statistical modeling of natural images. Journal of Mathematical Imaging and Vision 18, 17–33 (2003)
4. Heiler, M., Schnörr, C.: Natural image statistics for natural image segmentation. IJCV 63, 5–19 (2005)
5. Ruderman, D.: The statistics of natural images. Computation in Neural Systems 5, 517–548 (1994)

6. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV (2001)
7. Russell, B., Torralba, A.: Labelme: a database and web-based tool for image annotation. IJCV 77, 157–173 (2008)
8. Bileschi, S.: StreetScenes: Towards scene understanding in still images. Ph.D. Thesis, EECS, MIT (2006)
9. Rimey, R.D., Brown, C.M.: Control of selective perception using bayes nets and decision theory. IJCV 12, 173–207 (1994)
10. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for multi-class object layout. ICCV (2009)
11. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in context. ICCV (2007)
12. Gupta, A., Davis, L.S.: Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 16–29. Springer, Heidelberg (2008)
13. He, X., Zemel, R.S., Ray, D.: Learning and incorporating top-down cues in image segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 338–351. Springer, Heidelberg (2006)
14. Torralba, A.B.: Contextual priming for object detection. IJCV 53, 169–191 (2003)
15. Russell, B.C., Torralba, A.B., Liu, C., Fergus, R., Freeman, W.T.: Object recognition by scene alignment. NIPS (2007)
16. Heitz, G., Koller, D.: Learning spatial context: Using stuff to find things. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 30–43. Springer, Heidelberg (2008)
17. Vecera, S., Vogel, E., Woodman, G.: Lower region: A new cue for figure-ground assignment. *Journal of Experimental Psychology: General* 131, 194–205 (2002)
18. Fowlkes, C., Martin, D., Malik, J.: Local figureground cues are valid for natural images. *Journal of Vision* 7, 1–9 (2007)
19. Torralba, A.B., Oliva, A.: Depth estimation from image structure. *IEEE T-PAMI* 24, 1226–1238 (2002)
20. Hoiem, D., Efros, A.A., Hebert, M.: Recovering surface layout from an image. IJCV 75, 151–172 (2007)
21. Nedović, V., Smeulders, A., Redert, A.: Depth information by stage classification. In: ICCV (2007)
22. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. IJCV 42, 145–175 (2001)
23. (Supplimentary material)
24. Abbott, E.A.: *Flatland: A Romance of Many Dimensions* (1884)
25. Fischer, G., Nachtergaele, F., Prieler, S., van Velthuisen, H., Verelst, L., Wiberg, D.: *Global Agro-ecological Zones Assessment for Agriculture (GAEZ 2007)*. IIASA, Laxenburg, Austria and FAO, Rome, Italy (2007)
26. Lanitis, A., Taylor, C.J., Cootes, T.F.: Automatic interpretation and coding of face images using flexible models. *IEEE-PAMI* 19, 743–756 (1997)
27. Vogel, J., Schiele, B.: Semantic modeling of natural scenes for content-based image retrieval. IJCV 72, 133–157 (2007)
28. van de Sande, K., Gevers, T., Snoek, C.: Evaluating color descriptors for object and scene recognition. *IEEE T-PAMI* 99 (2009)
29. Vicente, S., Kolmogorov, V., Rother, C.: Graph cut based image segmentation with connectivity priors. In: CVPR (2008)

# Efficient Highly Over-Complete Sparse Coding Using a Mixture Model

Jianchao Yang<sup>1</sup>, Kai Yu<sup>2</sup>, and Thomas Huang<sup>1</sup>

<sup>1</sup> Beckman Institute, University of Illinois at Urbana Champaign, IL

<sup>2</sup> NEC Laboratories America, Cupertino, CA

{jyang29,huang}@ifp.illinois.edu, kyu@sv.nec-labs.com

**Abstract.** Sparse coding of sensory data has recently attracted notable attention in research of learning useful features from the unlabeled data. Empirical studies show that mapping the data into a significantly higher-dimensional space with sparse coding can lead to superior classification performance. However, computationally it is challenging to learn a set of highly over-complete dictionary bases and to encode the test data with the learned bases. In this paper, we describe a mixture sparse coding model that can produce high-dimensional sparse representations very efficiently. Besides the computational advantage, the model effectively encourages data that are similar to each other to enjoy similar sparse representations. What's more, the proposed model can be regarded as an approximation to the recently proposed local coordinate coding (LCC), which states that sparse coding can approximately learn the nonlinear manifold of the sensory data in a locally linear manner. Therefore, the feature learned by the mixture sparse coding model works pretty well with linear classifiers. We apply the proposed model to PASCAL VOC 2007 and 2009 datasets for the classification task, both achieving *state-of-the-art* performances.

**Keywords:** Sparse coding, highly over-complete dictionary training, mixture model, mixture sparse coding, image classification, PASCAL VOC challenge.

## 1 Introduction

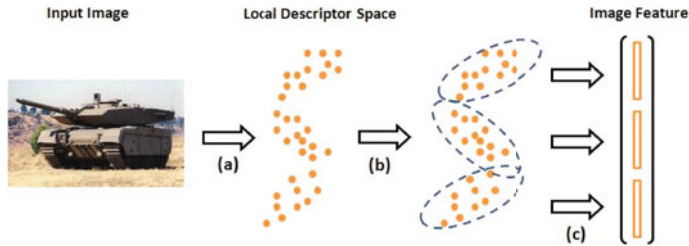
Sparse coding has recently attracted much attention in research of exploring the sparsity property in natural signals for various tasks. Originally applied to modeling the human vision cortex [1] [2], sparse coding approximates the input signal,  $\mathbf{x} \in R^d$ , in terms of a sparse linear combination of an over-complete bases or dictionary  $\mathbf{B} \in R^{d \times D}$ , where  $d < D$ . Among different ways of sparse coding, the one derived by  $\ell_1$  norm minimization attracts most popularity, due to its coding efficiency with linear programming, and also its relationship to the NP-hard  $\ell_0$  norm in compressive sensing [3]. The applications of sparse coding range from image restorations [4] [5], machine learning [6] [7] [8], to various computer vision tasks [9] [10] [11] [12]. Many efficient algorithms aiming to find such a sparse representation have been proposed in the past several years [13]. Several

empirical algorithms are also proposed to seek dictionaries which allow sparse representations of the signals [4] [13] [14].

Many recent works have been devoted to learning discriminative features via sparse coding. Wright *et al.* [10] cast the recognition problem as one of finding a sparse representation of the test image in terms of the training set as a whole, up to some sparse error due to occlusion. The algorithm utilizes the training set as the dictionary for sparse coding, limiting its scalability in handling large training sets. Learning a compact dictionary for sparse coding is thus of much interest [6] [15], and the sparse representations of the signals are used as the features trained later with generic classifiers, e.g., SVM. These sparse coding algorithms work directly on the objects, and are thus constrained to modeling only simple signals, e.g., aligned faces and digits. For general image classification, such as object recognition and scene categorization, the above sparse coding scheme will fail, i.e., it is computationally prohibitive and conceptually unsatisfactory to represent generic images with various spatial contents as sparse representations in the above way.

For generic image understanding, hierarchical models based on sparse coding applied to local parts or descriptors of the image are explored. Ranzato *et al.* [16] proposed a neural network for learning sparse representations for local patches. Raina *et al.* [17] described an approach using sparse coding applying to image patches for constructing image features. Both showed that sparse coding can capture higher-level features compared to the raw patches. Kavukcuoglu *et al.* [18] presented an architecture and a sparse coding algorithm that can efficiently learn locally-invariant feature descriptors. The descriptors learned by this sparse coding algorithm performs on a par with the carefully engineered SIFT descriptors as shown in their experiments. Inspired by the *Bag-of-Features* model and the *spatial pyramid matching* kernel [19] in image categorization, Yang *et al.* [11] proposed the ScSPM method where sparse coding is applied to local SIFT descriptors densely extracted from the image, and a spatial pyramid max pooling over the sparse codes is used to obtain the final image representation. As shown by Yu *et al.* [7], sparse coding is approximately a locally linear model, and thus the ScSPM method can achieve promising performance on various classification tasks with linear SVM. This architecture is further extended in [12], where the dictionary for sparse coding is trained with back-propagation to minimize the classification error.

The hierarchical model based on sparse coding in [11] [12] achieves very promising results on several benchmarks. Empirical studies show that using larger dictionary for sparse coding to map the data into higher dimensional space will generate superior classification performance. However, the computation of both training and testing for sparse coding can be prohibitively heavy if the dictionary is highly over-complete. Although nonlinear regressor can be applied for fast inference [18], the dictionary training is still computationally challenging. Motivated by the work in [7] that sparse coding should be local with respect to the dictionary, we propose an efficient sparse coding scheme with highly over-complete dictionaries using a mixture model. The model is derived



**Fig. 1.** A simplified schematic illustration of the image encoding process using the mixture sparse coding scheme. (a) local descriptor extraction; (b) mixture modeling in the descriptor space; (c) sparse coding and feature pooling. Within each mixture, a small dictionary for sparse coding can be applied, thus speeding up the coding process.

via a variational approach, and the coding speed can be improved approximately at the rate of the mixture number. Fig. 1 illustrates the simplified version of the image encoding process. The mixture modeling allows a much smaller dictionary for describing each mixture well, and thus the sparse coding computation can be effectively boosted.

The remainder of this paper is organized as follows: Section 2 talks about two closely related works and the motivations; Section 3 presents the proposed model and a practical algorithm for learning the model parameters; in Section 4, classification results on PASCAL VOC 2007 and 2009 datasets are reported and compared with the existing systems; and finally Section 5 concludes our paper with discussions and future work.

## 2 Related Works and Motivations

### 2.1 Sparse Coding for Image Classification

We review the ScSPM system for image classification using sparse coding proposed in [11]. Given a large collection of local descriptors randomly extracted from training images  $X = [x_1, x_2, \dots, x_N]$ , where  $x_i \in R^{d \times 1}$  is the  $i^{th}$  local descriptor in column manner and  $N$  is the total number of local descriptors selected, the ScSPM approach first concerns learning an over-complete dictionary  $B \in R^{d \times D}$  by

$$\begin{aligned} \min_{B, \{\alpha_i\}_i^N} \sum_i^N \|x_i - B\alpha_i\|_2^2 + \lambda \|\alpha_i\|_{\ell_1} \\ \text{s.t.} \quad \|B(m)\|_2^2 \leq 1, m = 1, 2, \dots, D, \end{aligned} \quad (1)$$

where  $\ell_1$ -norm is used for enforcing sparsity,  $\lambda$  is to balance the representation fidelity and sparsity of the solution, and  $B(m)$  is the  $m^{th}$  column of  $B$ . Denote  $A = [\alpha_1, \alpha_2, \dots, \alpha_N]$ , Eq. 1 is optimized by alternating between  $B$  and  $A$ . Fixing  $B$ ,  $A$  is found by linear programming; and fixing  $A$ , optimizing  $B$  is a quadratically constrained quadratic programming.



Given a set of local descriptors extracted from an image or a sub-region of the image  $S = [x_1, x_2, \dots, x_s]$ , we define the *set-level* feature over this collection of local descriptors in two steps:

1. *Sparse coding*. Convert each local descriptor into a sparse code with respect to the trained dictionary  $B$ :

$$\hat{A}_s = \min_A \|S - BA\|_2^2 + \lambda \|A\|_{\ell_1}, \quad (2)$$

2. *Max pooling*. The set-level feature is extracted by pooling the maximum absolute value of each row of  $\hat{A}_s$ :

$$\beta_s = \max(|\hat{A}_s|). \quad (3)$$

Note that  $\hat{A}_s$  contains the sparse codes as columns. Max pooling extracts the highest response in the collection of descriptors with respect to each dictionary atom, yielding a representation robust to translations within the image or its sub-regions.

To incorporate the spatial information of the local descriptors, spatial pyramid is employed to divide the image into different spatial sub-regions over different spatial scales [19]. Within each spatial sub-region, we collect its set of local descriptors and extract the corresponding *set-level* feature. The final *image-level* feature is constructed by concatenating all these *set-level* features.

## 2.2 Local Coordinate Coding

Yu *et al.* [7] proposed a local coordinate coding (LCC) method for nonlinear manifold learning in high dimensional space. LCC concerns learning a nonlinear function  $f(\mathbf{x})$  on a high dimensional sparse  $\mathbf{x} \in R^d$ . The idea is to approximate the nonlinear function by locally linear subspaces, to avoid the ‘‘curse of dimensionality’’. One main result of LCC is that the nonlinear function  $f(\mathbf{x})$  can be learned in a locally linear fashion as stated in the following lemma:

**Lemma 1 (Linearization).** *Let  $B \in R^{d \times D}$  be the set of anchor points on the manifold in  $R^d$ . Let  $f$  be an  $(a, b, p)$ -Lipschitz smooth function. We have for all  $\mathbf{x} \in R^d$ :*

$$\left| f(\mathbf{x}) - \sum_{m=1}^D \alpha(m) f(B(m)) \right| \leq a \|\mathbf{x} - \gamma(\mathbf{x})\|^2 + b \sum_{m=1}^D |\alpha(m)| \|B(m) - \gamma(\mathbf{x})\|^{1+p}$$

where  $B(m)$  is the  $m^{\text{th}}$  anchor points in  $B$ ,  $\gamma(\mathbf{x}) = \sum_{m=1}^D \alpha(m) B(m)$  is the approximation of  $\mathbf{x}$ , and we assume  $a, b \geq 0$  and  $p \in (0, 1]$ . Note that on the left hand side, a nonlinear function  $f(\mathbf{x})$  is approximated by a linear function  $\sum_{m=1}^D \alpha(m) f(B(m))$  with respect to the coding  $\alpha$ , where  $\{f(B(m))\}_{m=1}^D$  is the set of function values on the anchor points. The quality of this approximation is

bounded by the right hand side, which has two terms: the first term  $\|\mathbf{x} - \gamma(\mathbf{x})\|$  means  $\mathbf{x}$  should be close to its physical approximation  $\gamma(\mathbf{x})$ , and the second term means that the coding should be local. Minimizing the right hand side will ensure good approximation for the nonlinear function. Note that this minimization differs from the standard sparse coding in the regularization term, where a weighted  $\ell_1$  norm is employed to encourage localized coding. Nevertheless, as shown by the experiments in [7], in the high dimensional space with unit feature normalization, empirically the standard sparse coding well approximates the local coordinate coding for classification purposes.

### 2.3 Motivation

It should be easy to see that the ScSPM approach [11] works as an approximation to the LCC in modeling the manifold of the local descriptor space. If linear SVM is used, the nonlinear function values  $\{f(B(m))\}_{m=1}^D$  are simply determined by the weights of the classifier. The final classification score is thus an aggregation of these function values. The ScSPM model shows promising classification results on generic images with linear classifiers. Nevertheless, there are two limitations with the ScSPM framework:

1. Standard sparse coding does not include locality constraints explicitly, and thus may be inaccurate in modeling the manifold, especially when the dictionary is not big enough;
2. The computation of sparse coding increases to be unaffordable when a large dictionary is necessary to fit the nonlinear manifold well.

To make a concrete argument, we show the ScSPM computation time for encoding one image as well as the performance (in Average Precision) for dictionaries of different sizes on PASCAL VOC 2007 dataset [20], where 30,000 local descriptors are extracted from each image. As shown, the performance keeps growing as the dictionary size increases, as well as the computation time, which increases approximately linearly. In our experiment, training dictionaries beyond size 8192 is almost infeasible. The local coordinate coding (LCC) work suggests that the sparse coding should be local and the bases far away from the current encoding point can be discarded. This motivates our local sparse coding scheme induced by a Mixture Model, where local sparse coding within each mixture can be very fast (Refer to Fig. 1). For comparison, using 1024 mixtures with dictionary size 256 for each mixture, the effective dictionary size is  $1024 \times 256 = 262,144$ , and our proposed approach can process one image (with 30,000 local descriptors) in about one minute.

## 3 Sparse Coding Using a Mixture Model

The proposed approach partitions the descriptor space via a mixture model, where within each mixture a small over-complete dictionary is used to fit the local sub-manifold. An variational EM approach is applied to learn the model

**Table 1.** The relationships between the dictionary size and the computation time as well as the performance on PASCAL VOC 2007 validation dataset. The computation time reported is an approximate time needed for encoding one image.

Dictionary Size	512	2048	8192	32,768
Computation Time	1.5 mins	3.5 mins	14 mins	N/A
Performance	45.3%	50.2%	53.2%	N/A

parameters. Because of the descriptor space partition and dictionary sharing within each mixture, we can ensure that the sparse coding is local and similar descriptors have similar sparse codes. The image feature is finally constructed by pooling the sparse codes within each mixture.

### 3.1 The Model

We describe the image local descriptor space using a  $K$ -mixture model, where the local distribution of each mixture is further governed by an over-complete dictionary. Let  $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$  be the  $N$  independent and identically distributed observation points, and  $\mathbf{z} = \{z_n\}_{n=1}^N$  be the corresponding  $N$  hidden variables, where  $z_n \in \{1, 2, \dots, K\}$  is a random variable indicating the mixture assignments. Denote the mixture model parameters as  $\Theta = \{\mathbf{B}, \mathbf{w}\}$ , where  $\mathbf{B} = \{B_k\}_{k=1}^K$  is the set of over-complete dictionaries, where  $B_k \in \mathbb{R}^{d \times D}$ , and  $\mathbf{w} = \{w_k\}_{k=1}^K$  is the set of prior weights for the mixtures. We desire to learn the model by maximizing the likelihood

$$P(\mathbf{X}|\Theta) = \prod_{n=1}^N P(\mathbf{x}_n|\Theta) = \prod_{n=1}^N \sum_{z_n=1}^K w_{z_n} p(\mathbf{x}_n|B_{z_n}) \quad (4)$$

where we let

$$p(\mathbf{x}_n|B_{z_n}) = \int p(\mathbf{x}_n|B_{z_n}, \alpha_n^{z_n}) p(\alpha_n^{z_n}|\sigma) d\alpha_n \quad (5)$$

be the marginal distribution of a latent-variable model with a Laplacian prior  $p(\alpha_n^{z_n}|\sigma)$  on the latent variable  $\alpha_n^{z_n}$ , and  $p(\mathbf{x}_n|B_{z_n}, \alpha_n^{z_n})$  is modeled as a zero-mean isotropic Gaussian distribution regarding the representation error  $\mathbf{x}_n - B_{z_n} \alpha_n^{z_n}$ .

Learning the above model requires to compute the posterior  $P(\mathbf{z}|\mathbf{X}, \Theta)$ . However, under this model, this distribution is infeasible to compute in a closed form. Note that approximation can be used for the marginal distribution  $p(\mathbf{x}_n|B_{z_n})$  (introduced later in Eq. 9) in order to compute the posterior. This requires evaluating the mode of the posterior distribution of the latent variable for each data point, which, however, is computationally too slow. We thus develop a fast variational approach, where the posterior  $p(\mathbf{z}_n|\mathbf{x}_n, \Theta)$  is approximated by

$$q(\mathbf{z}_n = k|\mathbf{x}_i, \Lambda) = \frac{\mathbf{x}_n^T A_k \mathbf{x}_n + b_k^T \mathbf{x}_n + c_k}{\sum_{k'} \mathbf{x}_n^T A_{k'} \mathbf{x}_n + b_{k'}^T \mathbf{x}_n + c_{k'}} \quad (6)$$

where  $\Lambda = \{(A_k, b_k, c_k)\}$ ,  $A_k$  is a positive definite matrix,  $b_k$  is a vector, and  $c_k$  is a scalar. For computational convenience, we assume  $A_k$  to be diagonal.  $\Lambda$  is a set of free parameters, determining the mixture partition in the descriptor space. Then the learning problem can be formulated as

$$\min_{\Theta, \Lambda} \sum_{n=1}^N \sum_{z_n=1}^K [-q(z_n | \mathbf{x}_n, \Lambda) \log p(\mathbf{x}_n, z_n | \Theta) + q(z_n | \mathbf{x}_n, \Lambda) \log q(z_n | \mathbf{x}_n, \Lambda)] \quad (7)$$

which minimizes an upper bound of the negative log-likelihood  $-\sum_{i=1}^N \log p(\mathbf{x}_i | \Theta)$  of the model [21].

### 3.2 Learning Algorithm

The learning problem in Eq. 7 can be cast into a standard variational EM algorithm, where we optimize  $\Lambda$  to push down the upper bound to approximate the negative log-likelihood at E-step, and then update  $\Theta$  in the M-step to maximize the approximated likelihood. Let the first term in the object be formulated into

$$\begin{aligned} & \sum_{n=1}^N \sum_{z_n=1}^K g(z_n | \mathbf{x}_n, \Lambda) \log p(\mathbf{x}_n, z_n | \Theta) \\ &= \sum_{n=1}^N \sum_{z_n=1}^K g(z_n | \mathbf{x}_n, \Lambda) \log p(\mathbf{x}_n | B_{z_n}) + \sum_{n=1}^N \sum_{z_n=1}^K g(z_n | \mathbf{x}_n, \Lambda) \log w_{z_n} \end{aligned} \quad (8)$$

Note that the marginal distribution  $p(\mathbf{x}_n | B_{z_n})$  is difficult to evaluate due to the integration. We then simplify it by using the mode of the posterior distribution of  $\alpha_n$ :

$$\begin{aligned} -\log p(\mathbf{x}_n | B_{z_n}) &\approx \min_{\alpha_n^{z_n}} \{-\log p(\mathbf{x}_n | B_{z_n}, \alpha_n^{z_n}) - \log p(\alpha_n^{z_n} | \sigma)\} \\ &= \min_{\alpha_n^{z_n}} \|\mathbf{x}_n - B_{z_n} \alpha_n^{z_n}\|_2^2 + \lambda \|\alpha_n^{z_n}\|_1 \end{aligned} \quad (9)$$

which turns the integration into a standard sparse coding (or LASSO) problem. We then have the following updates rules for learning the model

1. Optimize  $\Lambda$

$$\min_{\Lambda} \sum_{n=1}^N \sum_{z_n=1}^K \{q(z_n | \mathbf{x}_n, \Lambda) [-\log p(\mathbf{x}_n | B_{z_n}) - \log w_{z_n} + \log q(z_n | \mathbf{x}_n, \Lambda)]\} \quad (10)$$

2. Optimize  $\mathbf{B}$

$$\min_{\mathbf{B}} - \sum_{n=1}^N \sum_{z_n=1}^K q(z_n | \mathbf{x}_n, \Lambda) \log p(\mathbf{x}_i | B_{z_n}) \quad (11)$$

where each column of the dictionaries  $\{B_k\}_{k=1}^K$  is constrained to be of unit  $\ell_2$  norm. The optimization is again a quadratically constrained quadratic programming problem, similar to the procedure of updating  $B$  in Eq. 11.

3. Optimize  $\mathbf{w}$ 

$$\begin{aligned} \min_{\mathbf{w}} & - \sum_{n=1}^N \sum_{z_n=1}^K q(z_n | \mathbf{x}_n, \Lambda) \log w_{z_n} \\ \text{s.t.} & \sum_{z_n=1}^K w_{z_n} = 1 \end{aligned} \quad (12)$$

which always leads to  $w_{z_n} = \frac{1}{N} \sum_{n=1}^N q(z_n | \mathbf{x}_n, \Lambda)$  using the Lagrange multiplier.

By alternatively optimizing over  $\Lambda$ ,  $\mathbf{B}$  and  $\mathbf{w}$ , we are guaranteed to find a local minimum for the problem of Eq. 7. Note that  $\mathbf{B} = [B_1, B_2, \dots, B_K] \in R^{d \times KD}$  is the effective highly over-complete dictionary ( $KD \gg d$ ) to learn for sparse coding. The above mixture sparse coding model leverages the learning complexity by training  $B_k$  ( $k = 1, 2, \dots, K$ ) separately and independently in Step 2 given the posteriors from Step 1. On the other hand, since we specify all the mixture dictionaries  $B_k$  to be of the same size, their fitting abilities for each data mixture will affect the mixture model parameters in Step 1, and thus the mixture weights in Step 3. Therefore, the above training procedure will efficiently learn the highly over-complete dictionary  $\mathbf{B}$ , while ensuring that the mixture dictionaries can fit each data mixture equally well 1.

## 3.3 Practical Implementation

The above iterative optimization procedures can be very fast with proper initialization for  $\Lambda$ ,  $\mathbf{B}$ , and  $\mathbf{w}$ . We propose to initialize the model parameters by the following:

1. Initialize  $\Lambda$  and  $\mathbf{w}$ : fit the data  $\mathbf{X}$  into a Gaussian Mixture Model (GMM) with  $K$  mixtures. The covariance matrix of each mixture is constrained to be diagonal for computational convenience.

$$p(\mathbf{X} | \mathbf{v}, \Sigma, \mathbf{w}) = \prod_{n=1}^N \sum_{k=1}^K v_k \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k). \quad (13)$$

The above Gaussian Mixture Model can be trained with standard EM algorithm. Initialize  $A_k$ ,  $b_k$ ,  $c_k$  and  $w_k$  with  $\Sigma_k^{-1}$ ,  $-2\Sigma_k^{-1}\mu_k$ ,  $\mu_k^T \Sigma_k^{-1} \mu_k$  and  $v_k$  respectively.

2. Initialize  $\mathbf{B}$ : Sample the data  $\mathbf{X}$  into  $K$  clusters  $\{\mathbf{X}_k\}_{k=1}^K$ , according to the posteriors of the data points calculated from the above GMM. Train the corresponding over-complete dictionaries  $\{B_k^0\}_{k=1}^K$  for those clusters using the procedure discussed for Eq. 11. Initialize  $\mathbf{B}$  with this trained set of dictionaries.

---

<sup>1</sup> In [22], a Gaussian mixture model is proposed for image classification. Instead of using Gaussian to model each mixture, we use sparse coding, which can capture the local nonlinearity.

### 3.4 Image Encoding

The proposed model can be regarded as a good approximation to the LCC theory [7]: i) the mixture clustering ensures the locality of the sparse coding; ii) and the highly over-complete dictionary provides sufficient anchor points for well approximation of the nonlinear manifold. Similar to case in Sec. 2.1, suppose we have a set of local descriptors  $S = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_S]$  extracted from an image or its sub-region, the *set-level* feature is defined on the latent variables (sparse codes)  $\{\alpha_n^{z_n}\}$ . Specifically, the local descriptors are first assigned to multiple mixtures according to the posteriors, and then the sparse codes are extracted with the corresponding dictionaries. We pool these sparse codes using a weighted average within each mixture and stack them into a super-vector:

$$f_s = [\sqrt{w_1}\mu_1^\alpha; \sqrt{w_2}\mu_2^\alpha; \dots; \sqrt{w_K}\mu_K^\alpha] \quad (14)$$

where

$$\mu_k^\alpha = \frac{\sum_{n=1}^N q(z_n = k|\mathbf{x}_n, \Lambda)\alpha_n^{z_n}}{\sum_{n=1}^N q(z_n = k|\mathbf{x}_n, \Lambda)} \quad (15)$$

is the weighted average of the sparse codes with their posteriors for the  $k^{th}$  mixture. The super-vector feature representation Eq. 14 has several characteristics that are not immediately obvious:

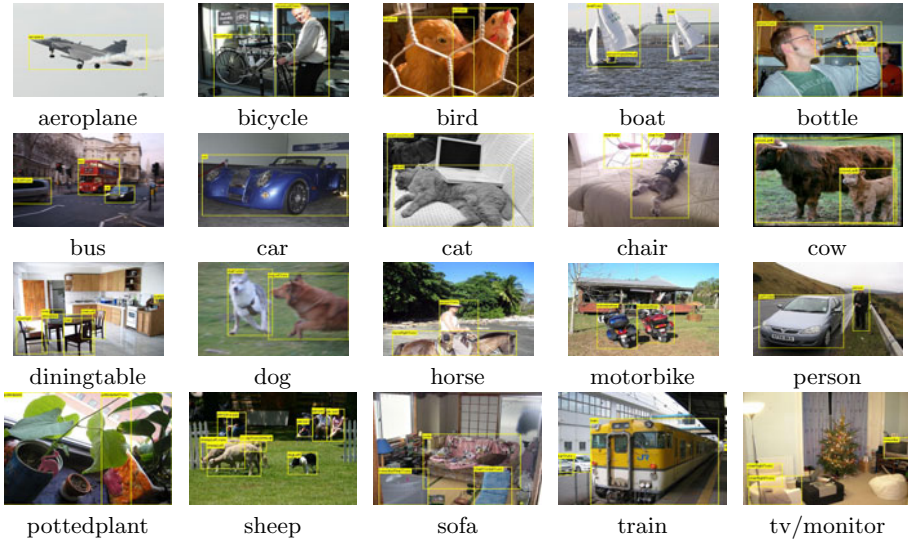
- The feature constructed in Eq. 14 is based on the locally linear model assumption, and thus is well fitted to linear kernels.
- The square root operator on each weight  $w_k$  corresponds to the linearity of the feature.
- In practice, the posteriors  $\{p(z_n = k|\mathbf{x}_n, \Lambda)\}_{k=1}^K$  are very sparse, i.e., each data point will be assigned to only one or two mixtures. Therefore, Eq. 15 is very fast to evaluate.
- The effective dictionary size of the sparse coding is  $K \times D$ . However, in our mixture sparse coding model, the nonlinear coding only involves dictionaries of size  $D$ , improving the computation approximately by  $K$  times (typically we choose  $K \geq 1024$ ).

Again, to incorporate the spatial information, we make use of the philosophy of spatial pyramid [19] to divide the image into multiple sub-regions over different configurations. The final image feature is then built by concatenating all the super-vectors extracted from these spatial sub-regions.

## 4 Experimental Validation

### 4.1 PASCAL Datasets

We evaluate the proposed model on the PASCAL Visual Object Classes Challenge (VOC) datasets. The goal of this challenge is to recognize objects from a number of visual object classes in realistic scenes (i.e., not pre-segmented objects). It is fundamentally a supervised learning problem in that a training set of labeled images is provided. Totally there are twenty object classes collected:



**Fig. 2.** Example images from Pascal VOC 2007 dataset

- **Person:** person
- **Animal:** bird, cat, cow, dog, horse, and sheep
- **Vehicle:** aeroplane, bicycle, boat, bus, car, motorbike, and train
- **Indoor:** bottle, chair, dining table, potted plant, sofa, and tv/monitor

Two main competitions for the PASCAL VOC challenge are organized:

- **Classification:** for each of the twenty classes, predicting presence/absence of an example of that class in the test image.
- **Detection:** predicting the bounding box and label of each object from the twenty target classes in the test image.

In this paper, we apply our model for the classification task to both PASCAL VOC Challenge 2007 and 2009 datasets.

The PASCAL VOC 2007 dataset [20] consists of 9,963 images, and PASCAL VOC 2009 [23] collects even more, 14,743 images in total. Both datasets are split into 50% for training/validation and 50% for testing. The distributions of images and objects by class are approximately equal across the training/validation and test sets. These images range between indoor and outdoor scenes, close-ups and landscapes, and strange viewpoints. These datasets are extremely challenging because all the images are daily photos obtained from Flickr where the size, viewing angle, illumination, etc appearances of the objects and their poses vary significantly, with frequent occlusions. Fig. 2 shows some example images for the twenty classes from PASCAL VOC 2007 dataset.

The classification performance is evaluated using the Average Precision (AP) measure, the standard metric used by PASCAL challenge, which computes the

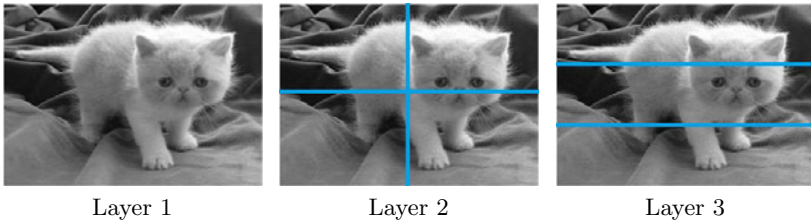
area under the Precision/Recall curve. The higher the score, the better the performance.

## 4.2 Implementation Details

**Local descriptor.** In our experiments, we only use single descriptor type HoG as the local descriptors, due to its computational advantage over SIFT via integral image. These descriptors are extracted from a regular grid with step size 4 pixels on the image plane. At each location, three scales of patches are used for calculating the HoG descriptor:  $16 \times 16$ ,  $24 \times 24$  and  $32 \times 32$ . As a result, approximately 30,000 local descriptors are extracted from each image. We then reduce the descriptor dimension from 128 to 80 with PCA.

**Mixture modeling.** For the VOC 07 dataset,  $K = 1024$  mixtures are used and the size of the dictionary  $D$  for each mixture is fixed to be 256. Therefore, the effective dictionary size is  $1024 \times 256 = 262144$ . Recall from Tab. 1 that working directly on a dictionary of this size is impossible. Using our mixture model, we only need to perform sparse coding on dictionaries of size 256, with little extra efforts of computing the posteriors for each descriptor, leveraging the computation time for encoding one image below a minute. For the VOC 09 dataset, we increase the mixture number to 2048.  $K$  and  $D$  are chosen empirically, balancing the performance and computational complexity.

**Spatial pyramid structure.** Spatial pyramid is employed to encode the spatial information of the local descriptors. As suggested by the winner system of VOC 2007 [24], we use the spatial pyramid structure shown in Fig. 3 for both datasets. Totally 8 spatial blocks are defined, and we extract a super-vector by Eq. 4 from each spatial block and concatenate them with equal weights.



**Fig. 3.** Spatial pyramid structure used in both PASCAL VOC 2007 and 2009 datasets

**Feature normalization.** Since our feature is based on the linear model assumption, we use Linear Discriminant Analysis (LDA) to sphere the features, and then linear SVM or Nearest Centroid is applied for classification. In practice, we always observe some improvements from this normalization step.



### 4.3 Classification Results

We present the classification results on the two datasets in this section. The precisions for each object class and the Average Precision (AP) are given by comprehensive comparisons.

**PASCAL VOC 2007 dataset.** For VOC 2007 dataset, the results we have are obtained by training on the training set and testing on the validation set. We report our results in Tab. 2 where the results of the winner system of VOC 2007 [24] and a recent proposed algorithm LLC [25] on validation set are also provided as reference. As the detailed results for Winner’07 and LLC are not available, we only cite their APs. Note that the Winner’07 system uses multiple descriptors beside dense SIFT, and the multiple kernel weights are also optimized for best performance. The LLC algorithm, similar to our system, only employs single kernel based on single descriptor. In both cases, our algorithm outperforms Winner’07 and LLC by a significant margin of about 5% in terms of AP.

**Table 2.** Image classification results on PASCAL VOC 2007 validation dataset

Obj. Class	aero	bicyc	bird	boat	bottle	bus	car	cat	chair	cow	
Winner’07	-	-	-	-	-	-	-	-	-	-	
LLC [25]	-	-	-	-	-	-	-	-	-	-	
Ours	78.5	61.6	53.0	69.8	31.69	62.2	81.0	60.5	55.9	41.8	
Obj. Class	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	AP
Winner’07	-	-	-	-	-	-	-	-	-	-	54.2
LLC [25]	-	-	-	-	-	-	-	-	-	-	55.1
Ours	59.3	50.3	75.4	72.9	82.1	26.1	36.1	55.7	81.6	56.3	59.6

**PASCAL VOC 2009 dataset.** Tab. 3 shows our results and comparisons with the top systems in VOC 2009. In this table, we compare with Winner’09 system (from NEC-UIUC team), and two honorable mention systems UVAS (from

**Table 3.** Image classification results on PASCAL VOC 2009 dataset. Our results are obtained based on single local descriptor without combining detection results.

Obj. Class	aero	bicyc	bird	boat	bottle	bus	car	cat	chair	cow	
Winner’09	88.0	68.6	67.9	72.9	44.2	79.5	72.5	70.8	59.5	53.6	
UVAS	84.7	63.9	66.1	67.3	37.9	74.1	63.2	64.0	57.1	46.2	
CVC	83.3	57.4	67.2	68.8	39.9	55.6	66.9	63.7	50.8	34.9	
Ours	87.7	67.8	68.1	71.1	39.1	78.5	70.6	70.7	57.4	51.7	
Obj. Class	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	AP
Winner’09	57.5	59.0	72.6	72.3	85.3	36.6	56.9	57.9	85.9	68.0	66.5
UVAS	54.7	53.5	68.1	70.6	85.2	38.5	47.2	49.3	83.2	68.1	62.1
CVC	47.2	47.3	67.7	66.8	88.8	40.2	46.6	49.4	79.4	71.5	59.7
Ours	53.3	59.2	71.6	70.6	84.0	30.9	51.7	55.9	85.9	66.7	64.6

University of Amsterdam and University of Surrey) and CVC (from Computer Vision Centre Barcelona ). The Winner'09 system obtains its results by combining the detection scores from object detector. The UVAS system employs multiple kernel learning over multiple descriptors. The CVC system not only makes use of the detection results, but also unites multiple descriptors. Yet, our algorithm performs close to the Winner'09 system, and improves by a notable margin over the honorable mention systems.

## 5 Conclusion and Future Work

This paper presents an efficient sparse coding algorithm with a mixture model, which can work with much larger dictionaries that often offer superior classification performances. The mixture model softly partitions the descriptor space into local sub-manifolds, where sparse coding with a much smaller dictionary can fast fit the data. Using 2048 mixtures, each with a dictionary of size 256, i.e, effective dictionary size is  $2048 \times 256 = 524,288$ , our model can process one image containing 30,000 descriptor in about 1 minutes, which is completely impossible for traditional sparse coding. Experiments on PASCAL VOC datasets demonstrate the effectiveness of the proposed approach. One interesting finding we have is that although our method maps each image into an exceptionally high dimension space, e.g., the image from VOC 2009 dataset is mapped to a  $2048 \times 256 \times 8 = 4,194,304$  dimensional space (spatial pyramid considered), we haven't observe any evidence of overfitting. This is possibly owing to the locally linear model assumption from LCC. Tighter connections with LCC will be investigated in the future, regarding the descriptor mixture modeling and the sparse codes pooling.

**Acknowledgments.** The main part of this work was done when the first author was a summer intern at NEC Laboratories America, Cupertino, CA. The work is also supported in part by the U.S. Army Research Laboratory and U.S. Army Research Office under grand number W911NF-09-1-0383.

## References

1. Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* 381 (1996)
2. Olshausen, B.A., Field, D.J.: Sparse coding with an overcomplete basis set: A strategy employed by v1?. *Vision Research* (1997)
3. Donoho, D.L.: For most large underdetermined systems of linear equations, the minimal  $\ell^1$ -norm solution is also the sparsest solution. *Comm. on Pure and Applied Math.* (2006)
4. Aharon, M., Elad, M., Bruckstein, A.: K-svd: an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing* (2006)
5. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution as sparse representation of raw image patches. In: *CVPR* (2008)
6. Bradley, D.M., Bagnell, J.A.: Differential sparse coding. In: *NIPS* (2008)

7. Yu, K., Zhang, T., Gong, Y.: Nonlinear learning using local coordinate coding. In: *Advances in Neural Information Processing Systems*, vol. 22 (2009)
8. Cheng, B., Yang, J., Yan, S., Huang, T.: Learning with  $\ell_1$  graph for image analysis. *IEEE Transactions on Image Processing* (2010)
9. Cevher, V., Sankaranarayanan, A., Duarte, M.F., Reddy, D., Baraniuk, R.G., Chellappa, R.: Compressive sensing for background subtraction. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part II. LNCS*, vol. 5303, pp. 155–168. Springer, Heidelberg (2008)
10. Wright, J., Yang, A., Ganesh, A., Satri, S., Ma, Y.: Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* (February 2009)
11. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009)
12. Yang, J., Yu, K., Huang, T.: Supervised translation-invariant sparse coding. In: *IEEE Conference on Computer Vision and Pattern Recognition* (2010)
13. Lee, H., Battle, A., Raina, R., Ng, A.Y.: Efficient sparse coding algorithms. In: *NIPS* (2006)
14. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research* 11, 19–60 (2010)
15. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Supervised dictionary learning. In: *NIPS* (2008)
16. Ranzato, M.A., Boureau, Y.L., LeCun, Y.: Sparse feature learning for deep belief networks. In: *NIPS* (2007)
17. Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.Y.: Self-taught learning: Transfer learning from unlabeled data
18. Kavukcuoglu, K., Ranzato, M., Fergus, R., LeCun, Y.: Learning invariant features through topographic filter maps. In: *IEEE Conference on Computer Vision and Pattern Recognition* (2009)
19. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: *IEEE Conference on Computer Vision and Pattern Recognition* (2006)
20. Everingham, M., Gool, L.V., Williams, C., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge, VOC 2007 Results (2007), <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>
21. Bishop, C.M.: *Pattern recognition and machine learning*. Springer, Heidelberg (2006)
22. Zhou, X., Cui, N., Li, Z., Liang, F., Huang, T.: Hierarchical gaussianization for image classification. In: *IEEE International Conference on Computer Vision (ICCV)* (2009)
23. Everingham, M., Gool, L.V., Williams, C., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge, VOC 2009 Results (2009), <http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html>
24. Marszalek, M., Schmid, C., Harzallah, H., van de Weijer, J.: Learning representations for visual object class recognition. In: *PASCAL Visual Object Class Challenge (VOC) workshop* (2007)
25. Wang, J., Yang, J., Yu, K., Lv, F.: Locality-constrained linear coding for image classification. In: *IEEE Conference on Computer Vision and Pattern Classification* (2010)

# Attribute-Based Transfer Learning for Object Categorization with Zero/One Training Example

Xiaodong Yu and Yiannis Aloimonos

University of Maryland, College Park, MD, USA  
xdyu@umiacs.umd.edu, yiannis@cs.umd.edu

**Abstract.** This paper studies the one-shot and zero-shot learning problems, where each object category has only one training example or has no training example at all. We approach this problem by transferring knowledge from known categories (a.k.a *source categories*) to new categories (a.k.a *target categories*) via object attributes. Object attributes are high level descriptions of object categories, such as color, texture, shape, etc. Since they represent common properties across different categories, they can be used to transfer knowledge from source categories to target categories effectively. Based on this insight, we propose an attribute-based transfer learning framework in this paper. We first build a generative attribute model to learn the probabilistic distributions of image features for each attribute, which we consider as attribute priors. These attribute priors can be used to (1) classify unseen images of target categories (zero-shot learning), or (2) facilitate learning classifiers for target categories when there is only one training examples per target category (one-shot learning). We demonstrate the effectiveness of the proposed approaches using the *Animal with Attributes* data set and show state-of-the-art performance in both zero-shot and one-shot learning tests.

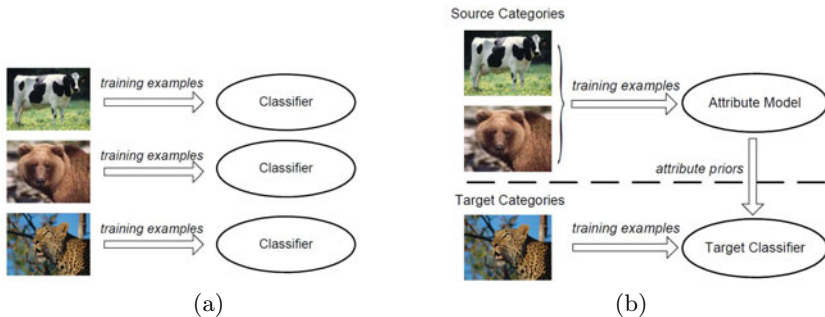
## 1 Introduction

In this paper, we focus on the one-shot learning [1] and the zero-shot learning [2] of object categories where there is only one training example per category or even no training example. Under these circumstances, conventional learning methods can not function due to the lack of training examples. To solve this problem, knowledge transfer becomes extremely important [3]: by transferring prior knowledge obtained from *source categories* (i.e. known categories) to *target categories* (i.e. unknown categories), we equivalently increase the number of training examples of the target categories. Thus, the difficulties raised by the scarcity of training examples can be greatly alleviated.

This paper present a transfer learning framework that utilizes the semantic knowledge of the object attributes. Object attributes are high-level descriptions about properties of object categories such as color, texture, shape, parts, context, etc. Human beings have a remarkable capability in recognizing unseen objects purely based on object attributes. For example, people who have never seen a zebra still could reliably identify an image of zebra if we tell them that “a zebra

is a wild quadrupedal with distinctive white and black strips living on African savannas”. Since they have prior knowledge about the related object attributes, e.g., *quadrupedal*, *white and black strips*, *African savannas*, they can transfer them to facilitate prediction of unseen categories. The attribute-based transfer learning framework is motivated by this insight. Figure 1 compares different learning process of conventional learning approaches and attribute-based transfer learning approaches: while conventional approaches treat each category individually and train each classifier from scratch, the attribute-based transfer learning approaches can help improve the learning of target classifiers using the attribute prior knowledge learned from source categories. Therefore, we are able to learn target classifiers with much fewer training examples, or even no examples. In the following, we will explore three key components in an attribute-based transfer learning system: attribute models, target classifiers and methods to transfer attribute priors. The main contributions of our paper are:

- 1) We present a generative attribute model that offers flexible representations for attribute knowledge transfer.
- 2) We propose two methods that effectively employ attribute priors in the learning of target classifiers and combine the training examples of target categories when they are available. Thus the attribute priors can help improving performance in both zero-shot and one-shot learning task.
- 3) We show state-of-the-art performance of our transfer learning system on the *Animal with Attributes* [2] data set.



**Fig. 1.** Comparison of the learning process between conventional learning approaches (a) and attribute-based transfer learning approaches (b)

The rest of this paper is organized as follows: Section 2 discusses the related work; Section 3 describes the attribute model, the target classifier and two approaches of knowledge transfer in details; we present the experimental results in Section 4 and conclude this paper in Section 5.

## 2 Related Work

Roughly, the methods of knowledge transfer for object categorization can be divided into three groups [3]: knowledge transfer by sharing either *features* [4,5],

*model parameters* [1,6] or *context information* [7]. Most of the early work relies on bootstrap approaches to select features or parameters to be transferred [4,5,1]. A very recent study [6] suggests that an explicit and controllable transfer of prior knowledge can be achieved by considering the ontological knowledge of object similarity. For example, *horse* and *giraffe* are both quadrupeds and share common topologies, so a full model can be transferred from horse to giraffe. The work presented in this paper integrates a broader ontological knowledge, i.e., object attributes, which can transfer knowledge either among similar categories (e.g., horse and giraffe), or among different categories that share common attributes (e.g., both German shepherd and giant panda have the attribute *black*).

Several recent studies have investigated the approach employing the object attributes in recognition problems [2,8,9,10]. Among them, our work is most related to [2,10]. However, as both studies focused on attribute prediction for zero-shot learning task, they did not attempt to combine attribute priors with the training examples of target categories. Thus, although useful, their applications in one-shot learning task are still limited. Since the framework presented in this paper (Figure 1b) includes the route for both attribute priors and the training examples of target categories, we can benefit from these two domains whichever is available in learning a new target category. Compared to the existing work in [2,10], our contribution is a more complete framework for attribute-based transfer learning, which enables us to handle both zero-shot learning and one-shot learning problems. The approaches in [8,9] are also related to ours. However, their methods need attributes annotated for each image. Although this type of image-level attribute annotation will benefit intra-class feature selection [8] and object localization [9], it requires substantially human efforts to label each image. Thus their scalability to a large number of categories is greatly restricted compared to the category-level attribute annotations advocated in [2,10] and this paper.

## 3 Algorithms

### 3.1 Background

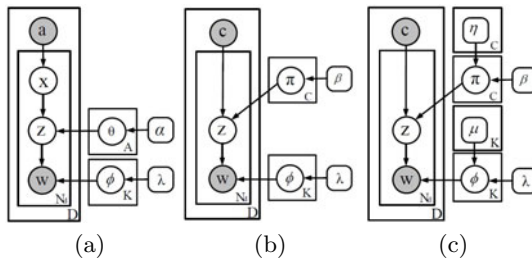
In the proposed approaches, the category-attribute relationship is represented by a category-attribute matrix  $\mathcal{M}$ , where the entry at the  $m$ -th row and the  $\ell$ -th column is a binary value indicating whether category  $m$  has the  $\ell$ -th attribute. Figure 3a illustrates an example of  $\mathcal{M}$ . Each object category thus has a list of attributes whose corresponding values in  $\mathcal{M}$  equal to “yes”. Given an object category, the list of associated attributes  $\mathbf{a}$  is deterministic. Take the category *cow* in Figure 3 for example, we have  $\mathbf{a} = \{black, white, brown, spots, furry, domestic\}$ . This information is supposed to be available for both source categories and target categories.

In our approach, the attribute model and the target classifier belong to an extension of topic models, which constitute an active research area in the machine learning community in recent years [11,12,13]. Computer vision researchers have extended them to deal with various vision problems [14,15,16,17]. In a topic

model, a document  $\mathbf{w}$  is modeled by a mixture of topics,  $z$ 's, and each topic  $z$  is represented by a probability distribution of words,  $w$ 's. In the computer vision domain, a quantized image feature is often analogous to a word (a.k.a “visual words” [14]), a group of co-occurred image features to a topic (a.k.a “theme” [17]), and an image to a document. In Section 4, we will visualize visual words and topics using examples in the test data set. In this paper, we use the bag-of-features image representation [18]: the spatial information of image features is discarded, and an image is represented as a collection of orderless visual words.

### 3.2 Attribute Model and Target Classifier

The attribute model we employed is the Author-Topic (AT) model (Figure 2.a) [13]. The AT model is originally designed to model the interests of authors from a given document corpus. In this paper, we extend the AT model to describe the distribution of image features related to attributes. To our best knowledge, this is the first attempt of this kind. Indeed, authors of a document and attributes of an object category have many similarities, which allow us to analogize the latter to the former: a document can have multiple authors and an object category can have multiple attributes; an author can write multiple documents and an attribute can be presented in multiple object categories. Nevertheless, there is also noticeable difference between them: each document can have a distinct list of authors, while all images within an object category share a common list of attributes.



**Fig. 2.** Graphical representations of the Author-Topic (AT) model (a), the Category-Topic (CT) model (b) and the CT model with informative Dirichlet priors over  $\pi$  and  $\phi$  (c). See text for detailed discussions of these models.

The AT model is a generative model. In this model, an image  $j$  has a list of attributes, denoted by  $\mathbf{a}_j$ . An attribute  $\ell$  in  $\mathbf{a}_j$  is modeled by a discrete distribution of  $K$  topics, which parameterized by a  $K$ -dim vector  $\theta_\ell = (\theta_{\ell 1}, \dots, \theta_{\ell K})$  with topic  $k$  receiving weight  $\theta_{\ell k}$ . The topic  $k$  is modeled by a discrete distribution of  $W$  codewords in the lexicon, which is parameterized by a  $W$ -dim vector  $\phi_k = (\phi_{k1}, \dots, \phi_{kW})$  with codeword  $v$  receiving weight  $\phi_{kv}$ . Symmetric Dirichlet priors are placed on  $\theta$  and  $\phi$ , with  $\theta_\ell \sim \text{Dirichlet}(\alpha)$ , and  $\phi_k \sim \text{Dirichlet}(\lambda)$ , where  $\alpha$  and  $\lambda$  are hyperparameters that affect the sparsity of these distributions. The generative process is outlined in Algorithm 1.

**Algorithm 1.** The generative process of the Author-Topic model

- 
- 1: given the attribute list  $\mathbf{a}_j$  and the desired number of visual words in image  $j$ ,  $N_j$
  - 2: **for**  $i = 1$  to  $N_j$  **do**
  - 3:   conditioning on  $\mathbf{a}_j$ , choose an attribute  $x_{ji} \sim \text{Uniform}(\mathbf{a}_j)$
  - 4:   conditioning on  $x_{ji}$ , choose a topic  $z_{ji} \sim \text{Discrete}(\theta_{x_{ji}})$ , where  $\theta_\ell$  defines the distribution of topics for attribute  $x = \ell$
  - 5:   conditioning on  $z_{ji}$ , choose a visual word  $w_{ji} \sim \text{Discrete}(\phi_{z_{ji}})$ , where  $\phi_k$  defines the distribution of visual words for topic  $z = k$
  - 6: **end for**
- 

Given a training corpus, the goal of inference in an AT model is to identify the values of  $\phi$  and  $\theta$ . In [13], Rosen-Zvi et al. presented a collapsed block Gibbs sampling method. The ‘‘collapse’’ means that the parameters  $\phi$  and  $\theta$  are analytically integrated out, and the ‘‘block’’ means that we draw the pair of  $(x_{ji}, z_{ji})$  together. The pair of  $(x_{ji}, z_{ji})$  is drawn according to the following conditional distribution

$$p(x_{ji} = \ell, z_{ji} = k | w_{ji} = v, \Omega) \propto \frac{\alpha/K + N_{\ell, \setminus ji}^k}{\alpha + \sum_{k'=1}^K N_{\ell, \setminus ji}^{k'}} \frac{\lambda/W + C_{k, \setminus ji}^v}{\lambda + \sum_{v'=1}^W C_{k, \setminus ji}^{v'}}, \quad (1)$$

where  $\Omega \equiv \{\mathbf{a}_j, \mathbf{z}_{\setminus ji}, \mathbf{x}_{\setminus ji}, \mathbf{w}_{\setminus ji}, \alpha, \lambda\}$ , the subscript  $ji$  represents the  $i$ -th visual word in image  $j$ ,  $x_{ji} = \ell$  and  $z_{ji} = k$  represent the assignments of current visual word to attribute  $\ell$  and topic  $k$  respectively,  $w_{ji} = v$  represents the observation that the current visual word is the  $v$ -th codeword in the lexicon,  $\mathbf{z}_{\setminus ji}$  and  $\mathbf{x}_{\setminus ji}$  represent all topic and attribute assignments in the training corpus excluding the current visual word,  $N_{\ell, \setminus ji}^k$  is the total number of visual words that are assigned to attribute  $\ell$  and topic  $k$ , excluding  $w_{ji}$ , and  $C_{k, \setminus ji}^v$  is the total number of visual words with value  $v$  that are assigned to topic  $k$ , excluding  $w_{ji}$ .

To run the Gibbs sampling algorithm, we first initialize  $\mathbf{x}$  and  $\mathbf{z}$  with random assignments. In each Gibbs sampling iteration, we draw samples of  $x_{ji}$  and  $z_{ji}$  for all visual words in the training corpus according to the distribution in Equation (1) in a randomly permuted order of  $i$  and  $j$ . The samples of  $\mathbf{x}$  and  $\mathbf{z}$  are recorded after the burn-in period. In experiments, we observe 200 iterations are sufficient for the sampler to be stable. The posterior means of  $\theta$  and  $\phi$  can then be estimated using the recorded samples as follows:

$$\hat{\theta}_{\ell k} = \frac{\alpha/K + N_{\ell}^k}{\alpha + \sum_{k'=1}^K N_{\ell}^{k'}}, \quad \hat{\phi}_{kv} = \frac{\lambda/W + C_k^v}{\lambda + \sum_{v'=1}^W C_k^{v'}}, \quad (2)$$

where  $N_{\ell}^k$  and  $C_k^v$  are defined in a similar fashion as in Equation (1), but without excluding the instance indexed by  $ji$ .

If there is only one attribute in each image and the attribute is the object category label, the AT model can be used in object categorization problems [16]. In this paper, we call this approach Category-Topic (CT) model (Figure 2.b) and use it as the target classifier in the proposed transfer learning framework.



It worth to note that the proposed transfer learning framework as illustrated in Figure 1b is an open framework in that we can also employ other type of attribute models and target classifiers. For example, we evaluate SVM as a target classifier in this paper. Nevertheless, our experiments show that the CT model can outperform discriminative classifiers such as SVM by a large margin.

The inference of a CT model can be performed in a similar way to the AT model. In the Gibbs sampling, we draw samples  $z_{ji}$  according to the following conditional distribution

$$p(z_{ji} = k | w_{ji} = v, c_j = m, \Omega) \propto \frac{\beta/K + M_{m, \setminus ji}^k}{\beta + \sum_{k'=1}^K M_{m, \setminus ji}^{k'}} \frac{\lambda/W + C_{k, \setminus ji}^v}{\lambda + \sum_{v'=1}^W C_{k, \setminus ji}^{v'}}, \quad (3)$$

where  $\Omega \equiv \{\mathbf{z}_{\setminus ji}, \mathbf{w}_{\setminus ji}, \beta, \lambda\}$ ,  $M_{m, \setminus ji}^k$  is the number of visual words in images of category  $m$  assigned to topic  $k$ , excluding the current instance. The posterior mean of  $\pi$  can be estimated as follows:

$$\hat{\pi}_{mk} = \frac{\beta/K + M_m^k}{\beta + \sum_{k'=1}^K M_m^{k'}}, \quad (4)$$

and the posterior mean of  $\phi$  is the same as in Equation (2).

After learning a CT model, we can use it to classify a test image  $\mathbf{w}_t = \{w_{t1}, \dots, w_{tN_t}\}$  by choosing the target classifier that yields the highest likelihood, where the likelihood for category  $c = m$  is estimated as

$$p(\mathbf{w}_t | c = m, \mathcal{D}^{\text{train}}) \approx \prod_{i=1}^{N_t} \sum_{k=1}^K \hat{\phi}_{kw_{ti}} \hat{\pi}_{mk}. \quad (5)$$

If the attribute list is unique in each category, an AT model can also be used to classify a new image by the maximum likelihood criterion. Suppose we have learned  $\theta_\ell$  for every  $\ell = 1, \dots, A$  from the source categories, we can then use them in classifying an image of a target category using the approximate likelihood

$$p(\mathbf{w}_t | c = m, \mathbf{a}_m, \mathcal{D}^{\text{train}}) \approx \prod_{i=1}^{N_t} \sum_{k=1}^K \hat{\phi}_{kw_{ti}} \left( \frac{1}{A_m} \sum_{\ell \in \mathbf{a}_m} \hat{\theta}_{\ell k} \right) \equiv \prod_{i=1}^{N_t} \sum_{k=1}^K \hat{\phi}_{kw_{ti}} \tilde{\pi}_{mk}, \quad (6)$$

where  $\mathbf{a}_m$  is the attribute list associated to a target category  $c = m$ ,  $A_m$  the length of  $\mathbf{a}_m$ . In the above equations, we have constructed a pseudo weight for the category-specified topic distribution of a new category from  $\hat{\theta}_\ell$ , i.e.,  $\tilde{\pi}_{mk} \equiv \left( \frac{1}{A_m} \sum_{\ell \in \mathbf{a}_m} \hat{\theta}_{\ell k} \right)$ . This pseudo weight can be viewed as the prior of  $\pi_m$  before we see the real training examples of the new category. Although the unique-attribute-list assumption does not hold in general, it is necessary for attribute-only classifiers, including the AT model discussed in this paper and the approaches in [28], to predict unseen categories. The data set tested in this paper satisfies this assumption.

While the AT model can be used to deal with the zero-shot learning problem, it is ineffective for the one-shot learning problem. One may conjecture to add the

training examples of target categories to those of source categories and then re-train the AT model. However, this naive approach will not work well in practice because the number of training examples of source categories is usually higher than the one of target categories by several orders. Consequently the AT model can not well represent the new observations in the training examples of target categories. Thus we need approaches to control the balance between the prior information from source categories and the new information in target categories. We will propose two approaches to achieve this goal in the rest of this section.

### 3.3 Knowledge Transfer by Synthesis of Training Examples

The first knowledge transfer approach is to synthesize training example for target categories. The idea is as follows: first, we learn the attribute model from the training examples of the source categories; second, for each target category, we run the generative process in Algorithm 1 to produce  $S$  synthesized training examples using the estimated  $\hat{\theta}$  and  $\hat{\phi}$  as well as the attribute list associated to this target category. Each synthesized training example contains  $\bar{N}$  visual words, where  $\bar{N}$  is the mean number of visual words per image in the source categories. In this procedure, the number of synthesized training example,  $S$ , represent our confidence about the attribute priors. We can use it to adjust the balance between the attribute priors and new observations from the training images of target categories.

Since we adopt the bag-of-features representation, the synthesized example is actually composed of a set of image features without spatial information. So they are indeed “artificial” examples in that we can not visualize them like a real image. This is different from the image synthesis approaches in the literature [19,20], which output viewable images. Nevertheless, since our goal is to generate training examples for the target categories to assist the learning process, this is not an issue providing the classifiers take these bag-of-features as inputs.

### 3.4 Knowledge Transfer by Informative Parameter Priors

The second knowledge transfer approach is to give parameters of the CT model in the target classifiers informative priors. Figure 2c illustrates the complete CT model, where  $\pi$  and  $\phi$  are given Dirichlet distributions as priors. In these Dirichlet distributions,  $\mu$  and  $\eta$  are base measurements that represent the mean of  $\phi$  and  $\pi$ , and  $\lambda$  and  $\beta$  are scaling parameters that control the sparsity of the samples drawn from the Dirichlet distribution. When we have no clue about the prior of  $\phi$  and  $\pi$ , we usually give symmetric Dirichlet priors, whose base measures are uniform distributions. The graphical representations of CT models often neglect such uniform distributed base measures and only retain the scaling parameters  $\lambda$  and  $\beta$ , as shown in Figure 2b. This rule also applies to the AT model. In this paper, these scaling parameters are given vague values when doing Gibbs sampling,  $\lambda = W$ ,  $\alpha = \beta = K$ .

However, after we learn the attribute model from source categories, our uncertainty about the  $\phi$  and  $\pi$  of target categories will be greatly reduced. Our

knowledge on these parameters are represented by the estimated  $\hat{\phi}$  in Equation (2) and  $\tilde{\pi}$  in Equation (6). Since  $E(\phi_k) = \mu_k$  and  $E(\pi_m) = \eta_m$ , now we can give informative priors to  $\phi$  and  $\pi$  by setting  $\mu_k = \hat{\phi}_k$  and  $\eta_m = \tilde{\pi}_m$ . The basic equation of Gibbs sampling of the CT model with informative prior the becomes

$$p(z_{ji} = k | w_{ji} = v, \Omega) \propto \frac{\beta \tilde{\pi}_{mk} + M_{m, \setminus ji}^k}{\beta + \sum_{k'=1}^K M_{m, \setminus ji}^{k'}} \frac{\lambda \hat{\phi}_{kv} + C_{k, \setminus ji}^v}{\lambda + \sum_{v'=1}^W C_{k, \setminus ji}^{v'}}, \quad (7)$$

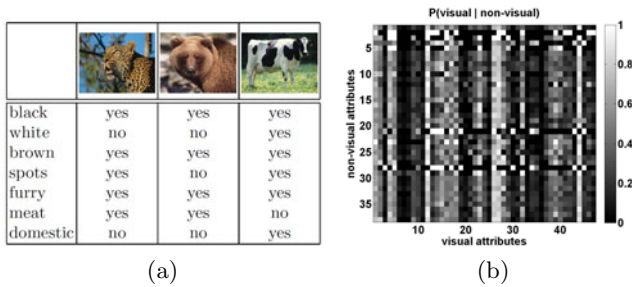
where  $\Omega \equiv \{c_j = m, \mathbf{z}_{\setminus ji}, \mathbf{w}_{\setminus ji}, \beta\eta, \lambda\mu\}$ . The posterior means of  $\pi$  and  $\phi$  in Equation (4) and (2) are updated accordingly. The value of  $\lambda$  and  $\beta$  represent our confidence on these priors, which can be used to control the balance between attribute priors and the new observations of training images of target categories. In the experiments, we set  $\lambda = \beta = \bar{N}S$ , where  $\bar{N}$  and  $S$  are defined as in Section 3.3.

By comparing Equation (7) and Equation (3), we can appreciate the importance of informative priors for the zero-shot learning task. If we have no prior knowledge about  $\pi$ , we can only give it a symmetric Dirichlet prior where  $\eta_{mk} = 1/K$ . In this scenario, the CT model have to see some training examples of target categories; otherwise,  $\pi_{mk}$  will be assigned to vague value  $1/K$ , which is useless for categorization tasks. Thus the CT model can not be used in zero-shot learning task. With the attribute knowledge, we can give  $\pi$  informative priors  $\eta_{mk} = \tilde{\pi}_{mk}$ , which permits us to perform zero-shot learning task using the CT model. Similar impact of the informative priors can be observed in the one-shot learning task.

## 4 Experiments

### 4.1 Data Set and Image Features

In the experiments, we use the ‘‘Animals with Attributes’’ (AwA) data set described in [2]. This data set includes 30475 images from 50 animal categories, and 85 attributes to describe these categories. The category-attribute relationship is labeled by human subjects and presented in a  $50 \times 85$  matrix  $\mathcal{M}$ . Figure 3.a illustrates a subset of this matrix. 40 categories are selected as source categories and the rest 10 categories are used as target categories. The division of source and target categories is the same as in [2]. The 85 attributes can be informally divided into two groups: visual attributes such as *black*, *furry*, *big*, *arctic*, etc., and non-visual attributes such as *fast*, *weak*, *fierce*, *domestic*, etc. Totally there are 38 non-visual attributes (attribute No.34 to No.64 and attribute No.79 to No.85) and 47 visual attributes. While non-visual attributes are not directly linked to visual features, it turns out that the non-visual attributes have strong correlation to the visual attributes, as shown in Figure 3.b. Take the attribute *fast* as an example, the top three most related visual attributes are *furry* ( $P(\text{furry}|\text{fast}) = 0.833$ ), *tail* ( $P(\text{tail}|\text{fast}) = 0.833$ ) and *ground* ( $P(\text{ground}|\text{fast}) = 0.786$ ).



**Fig. 3.** (a): examples of ontological knowledge represented by the binary category-attribute values; (b): the probability of nonvisual attributes conditioned on visual attributes measured by  $P(\text{visual}|\text{nonnon-visual}) \equiv N(\text{visual}, \text{nonnon-visual})/N(\text{nonnon-visual})$ , where  $N(\cdot)$  denote the number of categories that have the particular attributes in the given data set. Images and attributes are from the “Animals with Attributes” data set [2].

All images are resized such that the longest side has 300 pixels. From each image, we extract four types of image features: SIFT [21], rgSIFT [22], local color histogram and local self-similarity histogram (LSS) [23]. Then for each type of feature, we build a visual lexicon of size 1000 by applying K-means clustering algorithms over features from 250 images randomly selected from source categories. Codewords from four type of features are combined into a single lexicon with 4000 codewords. Features in all images are quantized into one of the codewords in this lexicon. On average, there are about 5000 features in each image. So we set  $\bar{N} = 5000$  in the approaches of attribute knowledge transfer in Section 3.3 and Section 3.4. In [2], color histogram (CH) and PHOG features are also extracted from 21 cells of a 3-level spatial pyramids. In our experiments, we did not use these features because the topic model can not discover sensible patterns of co-occurrence of CH/PHOG from the sparse 21 CH/PHOG features in each image.

## 4.2 Experiment Setup and Implementation Details

**Baseline Algorithms.** In the experiments, we use Direct Attribute Prediction (DAP) [2] and SVM as baselines in the zero/one-shot learning tasks.

The DAP is selected as a baseline because it is the state-of-the-art approach for zero-shot learning on the AWA data set. DAP uses a SVM classifier that is trained from source categories to predict the presence of each attribute in the images of target categories. Then the attribute predictions are combined into a category label prediction in an MAP formulation. The original DAP can only perform zero-shot learning. For one-shot learning, we use predicted attributes as features and choose a 1NN classifier following the idea in [8]. We call this classifier as “DAP+NN” in this paper.

When we use the synthesized training examples to transfer attribute knowledge, many existing classifiers can be used as the target classifier. We choose SVM as a baseline in this case, mainly because SVM is one of the state-of-the-art classifiers with bag-of-features image representation [24].

**Implementation Details.** The AT model has  $K_0 = 10$  unshared topics per attribute in all tests. When using synthesized training examples, the CT model has 100 topics; when using informative priors, the number of topics in the CT model is the same as the total topics in the AT model. The SVM in the target classifiers is implemented using the C-SVC in LIBSVM with a  $\chi^2$  kernel. The kernel bandwidth and the parameter  $C$  are obtained by cross-validation on a subset of the source categories.

**Evaluation Methodology.** In the zero-shot learning scenario, both AT and DAP are trained using the first 100 images of each source category. Then we use the AT model to generate  $S = \{10, 20, 100\}$  synthesized examples for each target category. The CT and SVM classifiers will be trained using these synthesized examples. We denote them as “CT+S” and “SVM+S” respectively in the reported results. Also we use the learned  $\hat{\phi}$  and  $\tilde{\pi}$  in the AT model as informative priors for the CT model as described in Section 3.4, where we set  $S = \{2, 5, 10\}$ . We denote it as “CT+P” in the reported results.

In the one-shot learning scenario, CT and SVM classifiers are trained with the synthesized training examples/informative priors obtained in the zero-shot learning test plus the first  $M = \{1, 5, 10\}$  images of each target category. The AT model is trained with the first 100 images of each source category plus the first  $M$  images of each target category. DAP+NN uses the attribute predictions of the first  $M$  images of each target category as training data points to classify new images of target categories based on the nearest neighbor criterion.

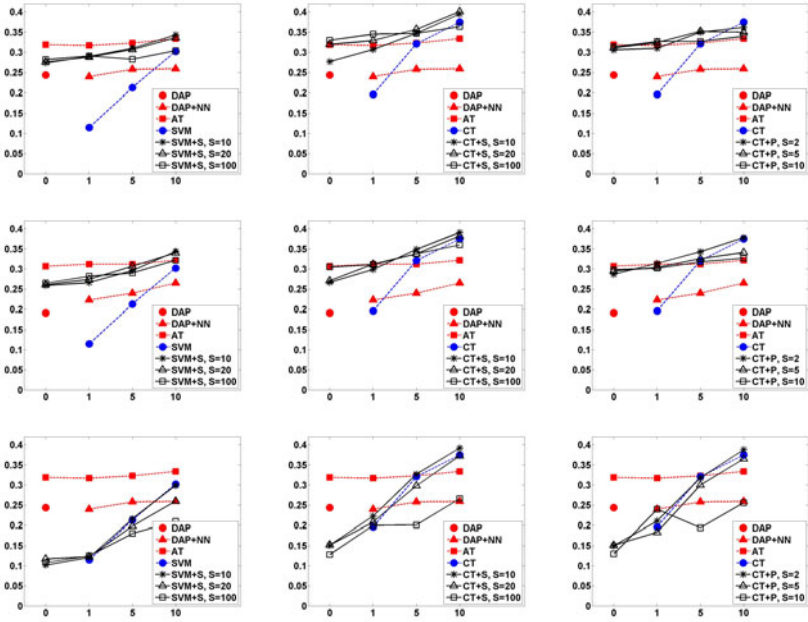
In both zero-shot and one-shot learning tests, all classifiers are tested over the last 100 images of each target category and the mean of the diagonal of the confusion matrix is reported as the measurement of performance.

### 4.3 Results

**Test 1: Overall Performance of Zero/One-Shot Learning.** The overall performance of zero/one-shot learning are presented in the top row of Figure 4. These results show that the proposed approach outperforms the baseline algorithms in the following three aspects:

1. *We have proposed a better attribute model for knowledge transfer.* In both zero/one-shot learning tests, the AT model surpasses DAP and DAP+NN by 5.9% to 7.9%. Furthermore, all target classifiers that employ the prior knowledge from the AT model (SVM+S, CT+S and CT+P) achieve higher accuracy than DAP and DAP+NN. These results clearly show the advantages of the AT model in the attribute-based transfer learning framework.

2. *We have proposed better methods of knowledge transfer for one-shot learning.* In the one-shot learning test, the performance of the AT model does not improve compared to the zero-shot learning test. It is not a surprise: there are total 4000 images of source categories while only 10 images of target categories in training the AT model, thus the learned AT model will be almost the same as the one trained only with the 4000 source images. This result shows that the naive method of knowledge transfer will not work for the one-shot learning



**Fig. 4.** Results of zero-shot and one-shot learning in Test 1 (top row, using all attributes), Test 2 (middle row, using visual attributes only) and Test 3 (bottom row, using randomly selected attributes) for SVM+S (column 1), CT+S (column 2) and CT+P (column 3) respectively. The x-axis represents the number of real examples,  $M$ , and the y-axis represents the mean classification accuracy, i.e., the mean of the diagonal of the confusion matrix.

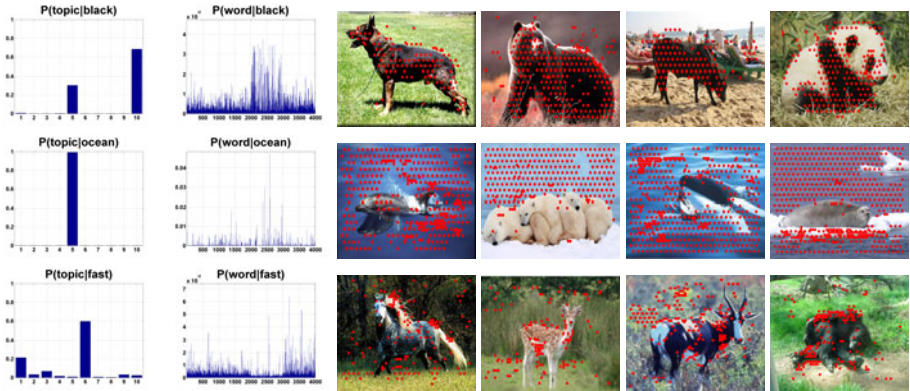
task. The proposed CT+S and CT+P approaches achieve better balance between the prior attribute knowledge and the real example of target categories, and the additional single training example improves their accuracies by 0.9%-3% (CT+S) and 0.4%-1.4% (CT+P) respectively compared to their zero-shot learning results.

3. *We have proposed a better target classifier.* In both the zero-shot and one-shot learning tasks, the CT models (CT+S and CT+P) consistently exceed the baseline SVM classifier and thus the advantage of CT over SVM in the zero/one-shot learning tasks is confirmed.

In addition to the above conclusion, we also have the following observations.

4. *CT+S generally outperforms CT+P.* CT+P can be viewed as an online version of CT+S, where the informative priors are equivalent to the initial values estimated from the synthesized examples in the initialization stage. Thus, samples drawn with CT+P are not distributed according to the true posterior distribution  $P(z_{ji}|z_{\setminus ji}, \mathbf{w})$ , which includes all the synthesized and real training examples. As a result, the categorization performance is degraded.

5. *With the increasing number of real training examples, the improvement on classification due to the prior knowledge decreases accordingly.* This suggests that



**Fig. 5.** Illustrations of three attribute models for *black*, *ocean* and *fast* from the top to the bottom. Column 1: the distribution of the 10 topics assigned to a particular attribute; Column 2: the distribution of codewords for a particular attribute; Column 3-6: examples of images from source categories (Column 3-5) and target categories (Column 6), superposed with the top 100 most likely codewords (solid red dots) for the attributes of the same row. Figures are best viewed in color.

the attributes do not contain all the information in target categories. Furthermore, some attributes may be difficult to learn and some are less informative to the categories. Thus when we have sufficient number of real training examples, the prior knowledge behaves more and more like noise and inevitably degrades the classification performance. We can thus derive a practical guideline from this observation to select an appropriate parameter  $S$ : when there is no or only one real training example, we can set a large value of  $S$ , e.g., 100; when more and more real training examples are available, we then gradually reduce the value of  $S$  to zero.

**Illustrations of the Attribute Models.** We show three attribute models for *black*, *ocean* and *fast* in Figure 5. Though we employ the bag-of-features image representation and discard the spatial information in the image representation, the visual features related to two visual attributes, *black* and *ocean*, roughly localize the regions of interest. As discussed in Section 4.1, the non-visual attribute, *fast*, is most correlated to visual attributes *furry*, *tail* and *ground*. So the visual features related to these visual attributes are implicitly linked to *fast*. Visual examples in Figure 5 support this assumption. The influence of the non-visual attributes on the classification performance will be evaluated quantitatively in Test 2.

**Test 2: The Influence of the Non-visual Attributes in the Transfer Learning.** In this experiment, we remove the non-visual attributes from the class-attribute matrix and repeat the above tests. Results are illustrated in the middle row of Figure 4. Clearly, the absence of non-visual attributes degrades the classification performance enormously for all classifiers in both zero-shot and

one-shot learning scenarios. This test illustrates the importance of the non-visual attributes in the transfer learning approaches.

**Test 3: The Effectiveness of the Knowledge of Attribute in the Transfer Learning.** In this experiment, we use the AT model learned from source categories to generate synthesized training examples or compute informative priors following **randomly selected** attributes for each target category, where the number of random attributes are the same as that of the true attributes in each target category. The results show that the classification performance is at the chance level in the zero-shot learning tasks. In the one-shot learning task, the prior knowledge from the randomly selected attributes does not improve the classification performance compared to those not using attribute priors. This experiment highlights the effectiveness of the knowledge of the attribute.

## 5 Conclusion and Future Work

In this paper, we proposed a transfer learning framework that employs object attributes to aid the learning of new categories with few training examples. We explore a generative model to describe the attribute-specified distributions of image features and two methods to transfer attribute priors from source categories to target categories. Experimental results show that the proposed approaches achieve state-of-the-art performance in both zero-shot and one-shot learning tests.

There are several areas to improve this work. First, we will evaluate our approaches using more data sets in the future, especially the FaceTracer data set [10] and the PASCAL+Yahoo data set [10]. We will also compare the attribute-based transfer learning approaches to those not using attributes, such as [4,5,11]. Second, we employ the bag-of-features image representation in this work, which discards valuable spatial information. In the future work, we will enhance the current model by including spatial constraints, such as regions [15] or vicinity [16]. By this way, we can localize attributes more accurately and subsequently improve the categorization performance. Finally, it would be highly valuable to formally study the influence of different visual attributes and select informative attributes for particular categories.

## Acknowledgement

The support of the Cognitive Systems program (under project POETICON) is gratefully acknowledged.

## References

1. Fei-Fei, L., Fergus, R., Perona, P.: One-Shot Learning of Object Categories. PAMI 28, 594–611 (2006)
2. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning To Detect Unseen Object Classes by Between-Class Attribute Transfer. In: CVPR (2009)



3. Fei-Fei, L.: Knowledge Transfer in Learning to Recognize Visual Object Classes. In: International Conference on Development and Learning (2006)
4. Bart, E., Ullman, S.: Cross-Generalization: Learning Novel Classes from a Single Example by Feature Replacement. In: CVPR, pp. 672–679 (2005)
5. Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing Features: Efficient Boosting Procedures for Multiclass Object Detection. In: CVPR, vol. 2, pp. 762–769 (2004)
6. Stark, M., Goesele, M., Schiele, B.: A Shape-Based Object Class Model for Knowledge Transfer. In: ICCV (2009)
7. Murphy, K., Torralba, A., Freeman, W.T.: Using the Forest to See the Trees: A Graphical Model Relating Features, Objects, and Scenes. In: NIPS (2003)
8. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing Objects by Their Attributes. In: CVPR (2009)
9. Wang, G., Forsyth, D.: Joint Learning of Visual Attributes, Object Classes and Visual Saliency. In: CVPR (2009)
10. Kumar, N., Belhumeur, P.N., Nayar, S.K.: FaceTracer: A Search Engine for Large Collections of Images with Faces. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 340–353. Springer, Heidelberg (2008)
11. Blei, D.M., Ng, A.Y., Jordan, M.I., Lafferty, J.: Latent Dirichlet Allocation. *JMLR* 3 (2003)
12. Griffiths, T.L., Steyvers, M.: Finding Scientific Topics. *Proceedings of the National Academy of Sciences* 101(suppl. 1), 5228–5235 (2004)
13. Rosen-Zvi, M., Chemudugunta, C., Smyth, P., Steyvers, M.: Learning author-topic models from text corpora. *ACM Transactions on Information System* (2009)
14. Sivic, J., Russell, B., Efros, A.A., Zisserman, A., Freeman, B.: Discovering Objects and Their Location in Images. In: ICCV, pp. 370–377 (2005)
15. 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: CVPR (2006)
16. Sudderth, E.B., Torralba, A., Freeman, W.T., Willsky, A.S.: Learning Hierarchical Models of Scenes, Objects, and Parts. In: ICCV, vol. 2, pp. 1331–1338 (2005)
17. Fei-Fei, L., Perona, P.: A Bayesian Hierarchical Model for Learning Natural Scene Categories. In: CVPR, pp. 524–531 (2005)
18. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual Categorization with Bags of Keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV (2004)
19. Sun, N., Haas, N., Connell, J.H., Pankanti, S.: A Model-Based Sampling and Sample Synthesis Method for Auto Identification in Computer Vision. In: IEEE Workshop on Automatic Identification Advanced Technologies, Washington, DC, USA, pp. 160–165 (2005)
20. Jiang, D., Hu, Y., Yan, S., Zhang, L., Zhang, H., Gao, W.: Efficient 3D reconstruction for face recognition. *Pattern Recognition* 38, 787–798 (2005)
21. Lowe, D.G.: Distinctive Image Features from Scale-invariant Keypoints. *IJCV* 20, 91–110 (2004)
22. van de Sande, K.E., Gevers, T., Snoek, C.G.: Evaluation of Color Descriptors for Object and Scene Recognition. In: CVPR (2008)
23. Shechtman, E., Irani, M.: Matching Local Self-Similarities across Images and Videos. In: CVPR, pp. 1–8 (2007)
24. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. *IJCV* 73, 213–238 (2007)

# Image Classification Using Super-Vector Coding of Local Image Descriptors

Xi Zhou<sup>1</sup>, Kai Yu<sup>2</sup>, Tong Zhang<sup>3</sup>, and Thomas S. Huang<sup>1</sup>

<sup>1</sup> Dept. of ECE, University of Illinois at Urbana-Champaign

<sup>2</sup> NEC Laboratories America, Cupertino, CA

<sup>3</sup> Department of Statistics, Rutgers University

**Abstract.** This paper introduces a new framework for image classification using local visual descriptors. The pipeline first performs a nonlinear feature transformation on descriptors, then aggregates the results together to form image-level representations, and finally applies a classification model. For all the three steps we suggest novel solutions which make our approach appealing in theory, more scalable in computation, and transparent in classification. Our experiments demonstrate that the proposed classification method achieves state-of-the-art accuracy on the well-known PASCAL benchmarks.

## 1 Introduction

Image classification, including object recognition and scene classification, remains to be a major challenge to the computer vision community. Perhaps one of the most significant developments in the last decade is the application of *local features* to image classification, including the introduction of “bag-of-visual-words” representation that inspires and initiates a lot of research efforts [1].

A large body of work investigates probabilistic generative models, with the objective towards understanding the semantic content of images. Typically those models extend the famous topic models on bag-of-word representation by further considering the spatial information of visual words [2, 3].

This paper follows another line of research on building discriminative models for classification. The previous work includes SVMs using pyramid matching kernels [4], biologically-inspired models [5, 6], and KNN methods [7, 8, 9]. Over the past years, the nonlinear SVM method using spatial pyramid matching (SPM) kernels [4, 10] seems to be dominant among the top performers in various image classification benchmarks, including Caltech-101 [11], PASCAL [12], and TRECVID. The recent improvements were often achieved by combining different types of local descriptors [10, 13, 14], without any fundamental change of the underlying classification method. In addition to the demand for more accurate classifiers, one has to develop more practical methods. Nonlinear SVMs scale at least quadratically to the size of training data, which makes it nontrivial to handle large-scale training data. It is thus necessary to design algorithms that are computationally more efficient.

## 1.1 Overview of Our Approach

Our work represents each image by a set of local descriptors with their spatial coordinates. The descriptor can be SIFT, or any other local features, computed from image patches at locations on a 2D grid. Our image classification method consists of three computational steps:

1. *Descriptor coding:*

Each descriptor of an image is nonlinearly mapped to form a high-dimensional sparse vector. We propose a novel nonlinear coding method called Super-Vector coding, which is algorithmically a simple extension of Vector Quantization (VQ) coding;

2. *Spatial pooling:*

For each local region, the codes of all the descriptors in it are aggregated to form a single vector, then vectors of different regions are concatenated to form the image-level feature vector. Our pooling is based on a novel probability kernel incorporating the similarity metric of local descriptors;

3. *Image classification:*

The image-level feature vector is normalized and fed into a classifier. We choose linear SVMs, which scale linearly to the size of training data.

We note that the *coding-pooling-classification* pipeline is the *de facto* framework for image scene classification. One notable example is the SPM kernel approach [4], which applies average pooling on top of VQ coding, plus a nonlinear SVM classifier using Chi-square or intersection kernels.

In this paper, we propose novel methods for each of the three steps and formalize their underlying mathematical principles. The work stresses the importance of learning good coding of local descriptors in the context of image classification, and makes the first attempt to formally incorporate the metric of local descriptors into distribution kernels. Putting all these together, the overall image classification framework enjoys a linear training complexity, and also a great interpretability that is missing in conventional models (see details in Sec. 2.3). The most importantly, our method demonstrates *state-of-the-art* performances on the challenging PASCAL07 and PASCAL09 image classification benchmarks.

## 2 The Method

In the following we will describe all the three steps of our image classification pipeline in detail.

### 2.1 Descriptor Coding

We introduce a novel coding method, which enjoys appealing theoretical properties. Suppose we are interested in learning a smooth nonlinear function  $f(x)$

defined on a high dimensional space  $\mathbb{R}^d$ . The question is, how to derive a good coding scheme (or nonlinear mapping)  $\phi(x)$  such that  $f(x)$  can be well approximated by a *linear function* on it, namely  $w^\top \phi(x)$ . Our only assumption here is that  $f(x)$  should be sufficiently smooth.

Let us consider a general unsupervised learning setting, where a set of bases  $C \subset \mathbb{R}^d$ , called *codebook* or *dictionary*, is employed to approximate any  $x$ , namely,

$$x \approx \sum_{v \in C} \gamma_v(x) v,$$

where  $\gamma(x) = [\gamma_v(x)]_{v \in C}$  is the coefficients, and sometimes  $\sum_v \gamma_v(x) = 1$ . By restricting the cardinality of nonzeros of  $\gamma(x)$  to be 1 and  $\gamma_v(x) \geq 0$ , we obtain the Vector Quantization (VQ) method

$$v_*(x) = \arg \min_{v \in C} \|x - v\|,$$

where  $\|\cdot\|$  is the Euclidean norm (2-norm). The VQ method uses the coding  $\gamma_v(x) = 1$  if  $v = v_*(x)$  and  $\gamma_v(x) = 0$  otherwise. We say that  $f(x)$  is  $\beta$  Lipschitz derivative smooth if for all  $x, x' \in \mathbb{R}^d$ :

$$|f(x) - f(x') - \nabla f(x')^\top (x - x')| \leq \frac{\beta}{2} \|x - x'\|^2.$$

It immediately implies the following simple function approximation bound via VQ coding: for all  $x \in \mathbb{R}^d$ :

$$\left| f(x) - f(v_*(x)) - \nabla f(v_*(x))^\top (x - v_*(x)) \right| \leq \frac{\beta}{2} \|x - v_*(x)\|^2. \quad (1)$$

This bounds simply states that one can approximate  $f(x)$  by  $f(v_*(x)) + \nabla f(v_*(x))^\top (x - v_*(x))$ , and the approximation error is upper bounded by the quality of VQ. It further suggests that the function approximation can be improved by learning the codebook  $C$  to minimize this upper bound. One way is the K-means algorithm

$$C = \arg \min_C \left\{ \sum_x \min_{v \in C} \|x - v\|^2 \right\}.$$

Eq. (1) also suggests that the approximation to  $f(x)$  can be expressed as a linear function on a nonlinear coding scheme

$$f(x) \approx g(x) \equiv w^\top \phi(x),$$

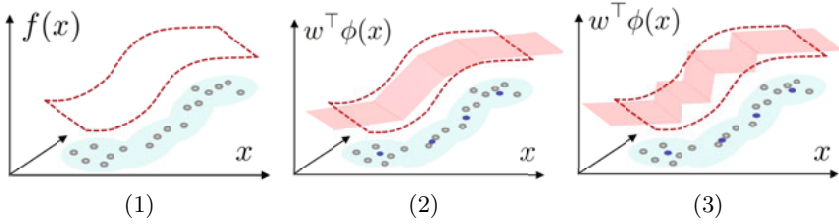
where  $\phi(x)$  is called the *Super-Vector* (SV) coding of  $x$ , defined by

$$\phi(x) = [s\gamma_v(x), \gamma_v(x)(x - v)^\top]_{v \in C}^\top \quad (2)$$

where  $s$  is a nonnegative constant. It is not difficult to see that  $w = [\frac{1}{s}f(v), \nabla f(v)]_{v \in C}$ , which can be regarded as unknown parameters to be estimated. Because  $\gamma_v(x) = 1$  if  $v = v_*(x)$ , otherwise  $\gamma_v(x) = 0$ , the obtained  $\phi(x)$

$a$  is highly sparse representation, with dimensionality  $|C|(d + 1)$ . For example, if  $|C| = 3$  and  $\gamma(x) = [0, 1, 0]$ , then

$$\phi(x) = \begin{bmatrix} \underbrace{0, \dots, 0}_{d+1 \text{ dim.}}, \underbrace{s, (x - v)^\top}_{d+1 \text{ dim.}}, \underbrace{0, \dots, 0}_{d+1 \text{ dim.}} \end{bmatrix}^\top \tag{3}$$



**Fig. 1.** Function  $f(x)$  approximated by  $w^\top \phi(x)$

As illustrated in Figure 1,  $w^\top \phi(x)$  provides a piece-wise linear function to approximate a nonlinear function  $f(x)$ , as shown in Figure 1(2), while with VQ coding  $\phi(x) = [\gamma_v(x)]_{v \in C}^\top$ , the same formulation  $w^\top \phi(x)$  gives a piece-wise constant approximation, as shown in Figure 1(3). This intuitively suggests that SV coding may achieve a lower function approximation error than VQ coding. We note that the popular bag-of-features image classification method essentially employs VQ to obtain histogram representations. The proposed SV coding is a simple extension of VQ, and may lead to a better approach to image classification.

## 2.2 Spatial Pooling

**Pooling.** Let each image be represented as a set of descriptor vectors  $x$  that follows an image-specific distribution, represented as a probability density function  $p(x)$  with respect to an image independent back-ground measure  $d\mu(x)$ . Let’s first ignore the spacial locations of  $x$ , and address the spacial pooling later. A kernel-based method for image classification is based on a kernel on the probability distributions over  $x \in \Omega$ ,  $K : \mathcal{P} \times \mathcal{P} \mapsto \mathbb{R}$ . A well-known example is the Bhattacharyya kernel [15]:

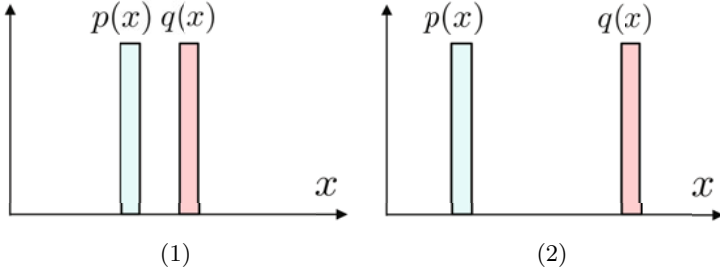
$$K_b(p, q) = \int_{\Omega} p(x)^{\frac{1}{2}} q(x)^{\frac{1}{2}} d\mu(x).$$

Here  $p(x)$  and  $q(x)$  represent two images as distributions over local descriptor vectors, and  $\mu(x)$  is the image independent background measure. Bhattacharyya kernel is closely associated with Hellinger distance, defined as  $D_h(p, q) = 2 - K_b(p, q)$ , which can be seen as a principled symmetric approximation of the

Kullback Leibler (KL) divergence [15]. Despite the popular application of both Bhattacharyya kernel and KL divergence, a significant drawback is the ignorance of the underlying similarity metric of  $x$ , as illustrated in Figure 2. In order to avoid this problem, one has to work with very smooth distribution families that are inconvenient to work with in practice. In this paper, we propose a novel formulation that explicitly takes the similarity of  $x$  into account:

$$\begin{aligned} K_s(p, q) &= \int_{\Omega} \int_{\Omega} p(x)^{\frac{1}{2}} q(x')^{\frac{1}{2}} \kappa(x, x') d\mu(x) d\mu(x') \\ &= \int_{\Omega} \int_{\Omega} p(x)^{-\frac{1}{2}} q(x')^{-\frac{1}{2}} \kappa(x, x') p(x) q(x') d\mu(x) d\mu(x') \end{aligned}$$

where  $\kappa(x, x')$  is a RKHS kernel on  $\Omega$  that reflects the similarity structure of  $x$ . In the extreme case where  $\kappa(x, x') = \delta(x - x')$  is the delta-function with respect to  $\mu(\cdot)$ , then the above kernel reduces to the Bhattacharyya kernel.



**Fig. 2.** Illustration of the drawback of Bhattacharyya kernel: in both cases their density kernels  $K_b(p, q)$  remain to be the same, equal to 0

In reality we cannot directly observe  $p(x)$  from any image, but a set  $X$  of local descriptors. Therefore, based on the empirical approximation to  $K_s(p, q)$ , we define a kernel between sets of vectors:

$$K(X, X') = \frac{1}{NN'} \sum_{x \in X} \sum_{x' \in X'} p(x)^{-\frac{1}{2}} q(x')^{-\frac{1}{2}} \kappa(x, x') \quad (4)$$

where  $N$  and  $N'$  are the sizes of the descriptor sets from two images.

Let  $\kappa(x, x') = \langle \phi(x), \phi(x') \rangle$ , where  $\phi(x)$  is the SV coding defined in the previous section. It is easy to see that  $\kappa(x, x') = 0$  if  $x$  and  $x'$  fall into different clusters. Then we have

$$K(X, X') = \frac{1}{NN'} \sum_{k=1}^{|C|} \sum_{x \in X_k} \sum_{x' \in X'_k} p(x)^{-\frac{1}{2}} q(x')^{-\frac{1}{2}} \kappa(x, x')$$

where  $X_k$  is the subset of  $X$  fallen into the  $k$ -th cluster. Furthermore, if we assume that  $p(x)$  remains constant within each cluster partition, i.e.,  $p(x)$  gives rise to a histogram  $[p_k]_{k=1}^{|C|}$ , then

$$K(X, X') = \frac{1}{NN'} \sum_{k=1}^{|C|} \left\langle \frac{1}{\sqrt{p_k}} \sum_{x \in X_k} \phi(x), \frac{1}{\sqrt{q_k}} \sum_{x' \in X'_k} \phi(x') \right\rangle$$

The above kernel can be re-written as an inner product kernel of the form  $K(X, X') = \langle \Phi(X), \Phi(X') \rangle$ , where

$$\Phi(X) = \frac{1}{N} \sum_{k=1}^{|C|} \frac{1}{\sqrt{p_k}} \sum_{x \in X_k} \phi(x).$$

Therefore functions in the reproducing kernel Hilbert space for this kernel has a linear representation  $f(X) = w^\top \Phi(X)$ . In other words, we can simply employ  $\Phi(X)$  as nonlinear feature vector and then learn a linear classifier using this feature vector. The effect is equivalent to using nonlinear kernel  $K(X, X')$  between image pairs  $X$  and  $X'$ .

Finally, we point out that weighting by histogram  $p_k$  is equivalent to treating density  $p(x)$  as piece-wise constant around each VQ basis, under a specific choice of background measure  $\mu(x)$  that equalizes different partitions. This representation is not sensitive to the choice of background measure  $\mu(x)$ , which is image independent. In particular, a change of measure  $\mu(\cdot)$  (still piece-wise constant in each partition) leads to a rescaling of different components in  $\Phi(X)$ . This means that the space of linear classifier  $f(x) = w^\top \Phi(X)$  remains the same.

**Spatial Pyramid Pooling.** To incorporate the spatial location information of  $x$ , we apply the idea of spatial pyramid matching [4]. Let each image be evenly partitioned into  $1 \times 1$ ,  $2 \times 2$ , and  $3 \times 1$  blocks, respectively in 3 different levels. Based on which block each descriptor comes from, the whole set  $X$  of an image is then organized into three levels of subsets:  $X_{11}^1, X_{11}^2, X_{12}^2, X_{21}^2, X_{22}^2, X_{11}^3, X_{12}^3$ , and  $X_{13}^3$ . Then we apply the pooling operation introduced in the last subsection to each of the subsets. An image's spacial pyramid representation is then obtained by concatenating the results of local pooling

$$\Phi_s(X) = \left[ \Phi(X_{11}^1), \Phi(X_{11}^2), \Phi(X_{12}^2), \Phi(X_{21}^2), \Phi(X_{22}^2), \Phi(X_{11}^3), \Phi(X_{12}^3), \Phi(X_{13}^3) \right]$$

### 2.3 Image Classification

Image classification is done by applying classifiers based on the image representations obtained from the pooling step. Here we consider the task of finding whether a particular category of objects is contained in an image or not, which can be translated into a binary classification problem. We apply a *linear* SVM that employs a hinge loss to learn  $g(X) = w^\top \Phi_s(X)$ . We note that the function is nonlinear on  $X$  since  $\Phi_s(X)$  is a nonlinear operator.

Interestingly, the image-level classification function is closely connected to a real-valued function on local descriptors. Without loss of generality, let's assume that only global pooling is used, which means  $\Phi_s(X) = \Phi(X)$  in this case.

$$g(X) = w^\top \Phi(X) = \frac{1}{N} \sum_{k=1}^{|C|} \frac{1}{\sqrt{p_k}} \sum_{x \in X_k} w^\top \phi(x) = \frac{1}{N} \sum_{k=1}^{|C|} \frac{1}{\sqrt{p_k}} \sum_{x \in X_k} g(x) \quad (5)$$

where  $g(x) = w^\top \phi(x)$ . The above equation provides an interesting insight to the classification process: a patch-level pattern matching is operated everywhere in the image, and the responses are then aggregated together to generate the score indicating how likely a particular category of objects is present. This observation is well-aligned with the biologically-inspired vision models, like Convolution Neural Networks [16] and HMAX model [6], which mostly employ feed-forward pattern matching for object recognition.

This connection stresses the importance of learning a good coding scheme on local descriptors  $x$ , because  $\phi(x)$  solely defines the function space of  $g(x) = w^\top \phi(x)$ , which consequently determines if the unknown classification function can be well learned. The connection also implies that supervised training of  $\phi(x)$  could potentially lead to further improvements.

Furthermore, the classification model enjoys the advantages of interpretability and computational scalability. Once the model is trained, Eq. (5) suggests that one can compute a response map based on  $g(x)$ , which visualizes where the classifier focuses on in the image, as shown in our experiments. Since our method naturally requires a linear classifier, it enjoys a training scalability which is linear to the number of training images, while nonlinear kernel-based methods suffer quadratic or higher complexity.

### 3 Discussion and Further Improvement

Our approach is along the line of recent works on unsupervised feature learning for image classification, especially, learning *sparse representations* e.g., [17, 5, 18, 19, 20]. In theory our work is more related to local coordinate coding (LCC) [19], which points out that in some cases a desired sparsity of  $\phi(x)$  should come from a *locality* of the coding scheme. Indeed, the proposed SV coding leads to a highly sparse representation  $\phi(x)$ , as defined by Eq. (2), which activates those coordinates associated to the neighborhood of  $x$ . As the result,  $g(x) = w^\top \phi(x)$  gives rise to a local linear function (i.e., piece-wise linear) to approximate the unknown nonlinear function  $f(x)$ . But, the computation of SV coding is much simpler than sparse coding approaches.

Our method can be further improved by considering a soft assignment of  $x$  to bases  $C$ . Recall that the underlying interpretation of  $f(x) \approx w^\top \phi(x)$  is the the approximation

$$f(x) \approx f(v_*(x)) + \nabla f(v_*(x))^\top (x - v_*(x))$$



which essentially uses the unknown function’s Taylor expansion at a nearby location  $v_*(x)$  to interpolate  $f(x)$ . One natural idea to improve this is using several neighbors in  $C$  instead of the nearest one. Let’s consider a soft K-means that computes  $p_k(x)$ , the posterior probability of cluster assignment for  $x$ . Then the function approximation can be handled as the expectation

$$f(x) \approx \sum_{k=1}^{|C|} p_k(x) \left[ f(v_k) + \nabla f(v_k)^\top (x - v_k) \right]$$

Then the pooling step becomes a computation of the expectation

$$\Phi(X) = \frac{1}{N} \left[ \frac{1}{\sqrt{p_k}} \sum_{x \in X} p_k(x) (x - v_k + s) \right]_{k=1}^{|C|}$$

where  $p_k = \frac{1}{N} \sum_{x \in X} p_k(x)$ , and  $s$  comes from Eq. (2). This approach is different from the image classification using GMM, e.g., [21][22]. Basically, those GMM methods consider the distribution kernel, while ours incorporates non-linear coding into the distribution kernel. Furthermore, our theory requires the stickiness to VQ – the soft version requires all the components share the same *isotropic* diagonal covariance. That means a much less number of parameters

**Table 1.** Comparison of different coding methods, on PASCAL VOC 2007 test set

AP (%)	VQ	GMM	SV	SV-soft
aeroplane	39.9	74.4	77.5	78.9
bicycle	44.0	57.9	67.2	68.4
bird	27.7	45.7	47.0	51.9
boat	53.8	68.9	73.9	71.5
bottle	15.8	26.2	27.2	29.8
bus	48.5	63.0	66.9	70.3
car	63.4	77.2	81.4	81.6
cat	38.6	54.6	61.1	60.2
chair	45.8	53.0	53.7	54.5
cow	27.4	42.7	49.3	48.2
dining_table	32.7	46.9	55.1	56.8
dog	36.0	43.1	44.6	44.9
horse	66.7	77.7	77.7	80.8
motorbike	43.6	60.2	66.2	68.8
person	73.1	83.6	84.8	85.9
potted_plant	25.9	28.2	28.5	29.6
sheep	22.8	42.3	46.7	47.7
sofa	41.9	51.2	56.1	57.7
train	60.0	75.6	79.2	81.7
tv/monitor	27.0	44.1	51.1	52.9
average	41.7	55.8	59.8	61.1

to estimate. Our experiment confirms that our approach leads to a significantly higher accuracy.

## 4 Experiments

We perform image classification experiments on two datasets: PASCAL VOC 2007 and PASCAL VOC 2009. The images in both datasets contain objects from 20 object categories and range between indoor and outdoor scenes, close-ups and landscapes, and strange viewpoints. The datasets are extremely challenging because of significant variations of appearances and poses with frequent occlusions. PASCAL VOC 2007 consists of 9,963 images which are divided into three subsets: training data (2501 images), validation data (2510 images), and test data (4952 images). PASCAL VOC 2009 consists of 14,743 images and correspondingly are divided into three subsets: training data(3473 images), validation data(3581 images), and testing data (7689 images).

All the following experiment results are obtained on the testing datasets, except the comparison experiment for different codebook sizes  $|C|$  (Table 4), which is performed on PASCAL VOC 2007 validation set. We use the PASCAL toolkit to evaluate the classification accuracy, measured by average precision based on the precision/recall curve.

**Table 2.** Comparison of our method with top performers in PASCAL VOC 2007

AP (%)	QMUL	TKK	XRCE	INRIA(flat)	INRIA(GA)	Ours
aeroplane	71.6	71.4	72.3	74.8	77.5	<b>79.4</b>
bicycle	55.0	51.7	57.5	62.5	63.6	<b>72.5</b>
bird	41.1	48.5	53.2	51.2	<b>56.1</b>	55.6
boat	65.5	63.4	68.9	69.4	71.9	<b>73.8</b>
bottle	27.2	27.3	28.5	29.2	33.1	<b>34.0</b>
bus	51.1	49.9	57.5	60.4	60.6	<b>72.4</b>
car	72.2	70.1	75.4	76.3	78.0	<b>83.4</b>
cat	55.1	51.2	50.3	57.6	58.8	<b>63.6</b>
chair	47.4	51.7	52.2	53.1	53.5	<b>56.6</b>
cow	35.9	32.3	39.0	41.1	42.6	<b>52.8</b>
dining_table	37.4	46.3	46.8	54.9	54.9	<b>63.2</b>
dog	41.5	41.5	45.3	42.8	45.8	<b>49.5</b>
horse	71.5	72.6	75.7	76.5	77.5	<b>80.9</b>
motorbike	57.9	60.2	58.5	62.3	64.0	<b>71.9</b>
person	80.8	82.2	84.0	84.5	<b>85.9</b>	85.1
potted_plant	15.6	31.7	32.6	36.3	36.3	<b>36.4</b>
sheep	33.3	30.1	39.7	41.3	44.7	<b>46.5</b>
sofa	41.9	39.2	50.9	50.1	50.6	<b>59.8</b>
train	76.5	71.1	75.1	77.6	79.2	<b>83.3</b>
tv/monitor	45.9	41.0	49.5	49.3	53.2	<b>58.9</b>
average	51.2	51.7	55.6	57.5	59.4	<b>64.0</b>

**Table 3.** Comparison of our method with top performers in PASCAL VOC 2009

AP (%)	LEOBEN	LIP6	LEAR	FIRSTNIKON	CVC	UVASURREY	OURS
aeroplane	79.5	80.9	79.5	83.3	86.3	84.7	<b>87.1</b>
bicycle	52.1	52.3	55.5	59.3	60.7	63.9	<b>67.4</b>
bird	57.2	53.8	54.5	62.7	<b>66.4</b>	66.1	65.8
boat	59.9	60.8	63.9	65.3	65.3	67.3	<b>72.3</b>
bottle	29.3	29.1	<b>43.7</b>	30.2	41.0	37.9	40.9
bus	63.5	66.2	70.3	71.6	71.7	74.1	<b>78.3</b>
car	55.1	53.4	66.4	58.2	64.7	63.2	<b>69.7</b>
cat	53.9	55.9	56.5	62.2	63.9	64.0	<b>69.7</b>
chair	51.1	50.7	54.4	54.3	55.5	57.1	<b>58.5</b>
cow	31.3	33.8	38.8	40.7	40.1	46.2	<b>50.1</b>
dining_table	42.9	43.9	44.1	49.2	51.3	54.7	<b>55.1</b>
dog	44.1	44.6	46.2	50.0	45.9	53.5	<b>56.3</b>
horse	54.8	59.4	58.5	66.6	65.2	68.1	<b>71.8</b>
motorbike	58.4	58	64.2	62.9	68.9	70.6	<b>70.8</b>
person	81.1	80.0	82.2	83.3	85.0	<b>85.2</b>	84.1
potted_plant	30.0	25.3	39.1	34.2	<b>40.8</b>	38.5	31.4
sheep	40.2	41.9	41.3	48.2	49	47.2	<b>51.5</b>
sofa	44.2	42.5	39.8	46.1	49.1	49.3	<b>55.1</b>
train	74.9	78.4	73.6	83.4	81.8	83.2	<b>84.7</b>
tv/monitor	58.2	60.1	66.2	65.5	<b>68.6</b>	68.1	65.2
average	53.1	53.6	56.9	58.9	61.1	62.1	<b>64.3</b>

In all the experiments, 128-dimensional SIFT vectors are extracted over a grid with spacing of 4 pixels on three patch scales (16x16, 25x25 and 31x31). The dimension of descriptors is reduced to 80 by applying principal component analysis (PCA). The codebooks  $C$  are trained on one million randomly sampled descriptors. The constant  $s$  is chosen from  $[0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}]$  via cross-validation on the training set.

#### 4.1 Comparison of Nonlinear Coding Methods

Our first experiment investigates image classification using various nonlinear coding methods. The goal is to study which coding method performs the best under linear SVM classifiers. These methods are: (1) VQ coding – using Bhattacharyya kernel on spatial pyramid histogram presentations; (2) GMM – the method described in [22]; (3) SV – the super-vector coding proposed by this paper; (4) SV-soft – the soft version of SV coding, where  $[p_k(x)]_k$  for each  $x$  is truncated to retain the top 20 elements with the rest elements being set zero.

Table 1 shows the experiment results with different coding methods on PASCAL VOC 2007 test dataset. In all the cases  $|C| = 512$  bases/components are used for coding. SV and SV-soft both significantly outperform other two competitors. SV-soft is slightly better than SV. In the rest of the experiments we apply SV-soft for classification.

**Table 4.** The influence of codebook sizes  $|C|$ , on PASCAL VOC 2007 validation set

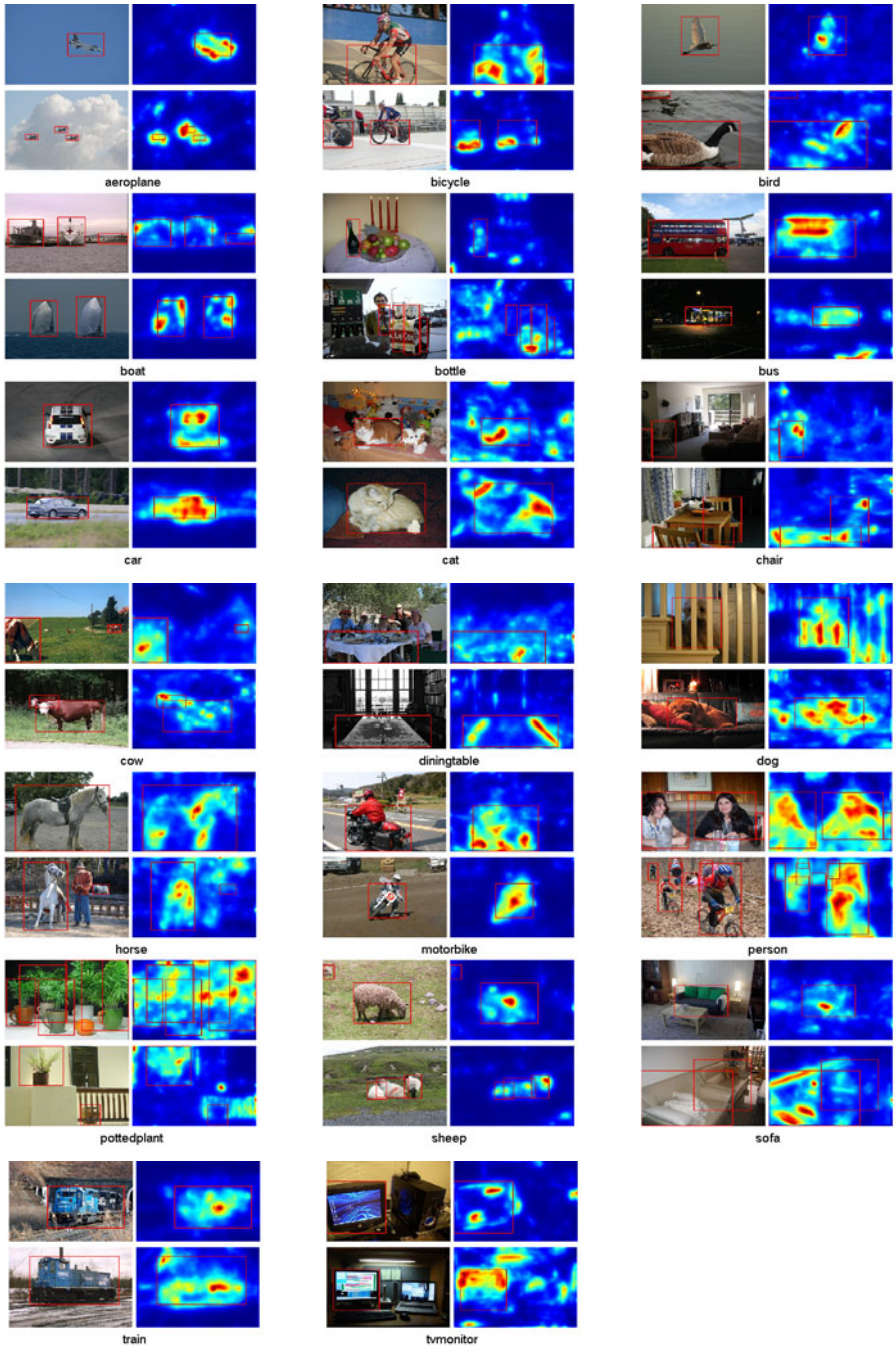
AP (%)	$ C  = 256$	$ C  = 512$	$ C  = 1024$	$ C  = 2048$
aeroplane	77.7	77.9	77.9	78.7
bicycle	55.6	57.2	58.2	58.7
bird	51.0	53.5	54.4	54.0
boat	66.3	66.9	67.1	68.9
bottle	25.5	29.8	31.5	31.9
bus	56.2	59.7	60.9	60.0
car	78.8	79.6	79.8	80.5
cat	59.5	61.4	62.3	62.4
chair	56.4	56.6	56.8	58.0
cow	40.0	43.6	45.6	44.3
dining_table	52.7	58.8	61.1	60.7
dog	42.3	46.5	48.7	47.1
horse	72.5	72.1	72.2	74.4
motorbike	65.7	68.7	70.1	70.5
person	79.8	81.0	81.6	81.7
potted_plant	23.3	22.9	22.5	23.2
sheep	30.2	33.9	35.5	32.0
sofa	52.2	54.7	55.9	57.3
train	80.2	81.2	81.4	82.5
tv/monitor	55.0	56.4	57.2	57.9
average	56.0	58.1	59.0	59.2

## 4.2 Comparison with State-of-the-Art Results

In this section we compare the performance of our method with reported state-of-the-art results on the PASCAL VOC 2007 and 2009 benchmarks. In both cases, we train the classifier on the training set plus the validation set, and evaluate on the test set, with  $|C|$  fixed as 2048. Table 2 compares the experiment results by our approach with the top performances in PASCAL VOC 2007 dataset,<sup>1</sup> while Table 3 compares our results with the top results in PASCAL VOC 2009 dataset.<sup>2</sup> In both cases, our method significantly outperforms the competing methods on most of the object categories. We note that most of those compared methods extend the SPM nonlinear SVM classifier by combing multiple visual descriptors/kernels, while our method utilizes only SIFT features on gray images. This difference highlights the significant success of the proposed approach. Note that in Table 3 we do not compare with the winner team NEC-UIUC's result, because as far as we know, they combined an object detection model, i.e. using the information of the provided bounding boxes, to achieve a higher accuracy.

<sup>1</sup> [http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/workshop/everingham\\_cls.pdf](http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/workshop/everingham_cls.pdf)

<sup>2</sup> [http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2009/workshop/everingham\\_cls.pdf](http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2009/workshop/everingham_cls.pdf)



**Fig. 3.** Visualization of the learned patch-level function  $g(x)$  on image examples from PASCAL-09. The relationship between  $g(x)$  and the image classification function  $g(X)$  is shown in Eq. 5. The figures show that  $g(x)$  has a good potential for object detection.

### 4.3 Impact of Codebook Size

In this section we report further experimental results on PASCAL VOC 2007 validation set, to show the impact of codebook size  $|C|$  on classification performance. As shown in Table 4, as we increase  $|C|$  from 256, to 512, 1024, and 2048, the classification accuracy keeps being improved. But the improvement gets small after  $|C|$  goes over 1024.

### 4.4 Visualization of the Learned Patch-Level Function

As suggested by Eq. 5, a very unique perspective of our method is the “transparency” of the classification model. Once the image classifier is trained, a real-valued function  $g(x)$  is automatically obtained on the local descriptor level. Therefore a response map of  $g(x)$  can be visualized on test images. In Figure 3, we show the response map (with kernel smoothing) on a set of random images from the PASCAL VOC 2009 test set. In most of the cases, the results are quite meaningful – the target objects are mostly covered by high-valued responses of  $g(x)$ . This observation suggests a potential to extend the current framework toward joint classification and detection.

## 5 Conclusion

This paper introduces a new method for image classification. The method follows the usual pipeline but introduces significantly novel methods for each of the steps. We formalizes the underlying mathematic principles for our methods and stresses the importance of learning a good coding of local descriptors in image classification. Compared to popular state-of-the-art methods, our approach is appealing in theory, more scalable in computation, transparent in classification, and produces state-of-the-art accuracy on the well-known PASCAL benchmark.

**Acknowledgments.** The main part of this work was done when the first author was a summer intern at NEC Laboratories America, Cupertino, CA.

## References

1. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV, vol. 1, p. 22 (2004) (Citeseer)
2. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories (2005) (Citeseer)
3. Sivic, J., Russell, B., Efros, A., Zisserman, A., Freeman, W.: Discovering object categories in image collections. In: Proc. ICCV, vol. 2 (2005)
4. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories (2006) (Citeseer)
5. MarcAurelio Ranzato, F., Boureau, Y., LeCun, Y.: Unsupervised learning of invariant feature hierarchies with applications to object recognition. In: Proc. Computer Vision and Pattern Recognition Conference (CVPR 2007) (2007) (Citeseer)

6. Serre, T., Wolf, L., Poggio, T.: Object recognition with features inspired by visual cortex. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, p. 994 (2005) (Citeseer)
7. Zhang, H., Berg, A., Maire, M., Malik, J.: SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In: Proc. CVPR, vol. 2, pp. 2126–2136 (2006) (Citeseer)
8. Makadia, A., Pavlovic, V., Kumar, S.: A new baseline for image annotation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 316–329. Springer, Heidelberg (2008)
9. Torralba, A., Fergus, R., Weiss, Y.: Small codes and large image databases for recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8 (2008)
10. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: Proceedings of the 6th ACM international conference on Image and video retrieval, p. 408. ACM, New York (2007)
11. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding* 106, 59–70 (2007)
12. Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision* (2009)
13. Varma, M., Ray, D.: Learning the discriminative power-invariance trade-off. In: Proc. ICCV, vol. 2007 (2007) (Citeseer)
14. Marszalek, M., Schmid, C., Harzallah, H., Weijer, J.V.D.: Learning object representations for visual object class recognition. In: Visual Recognition Challenge workshop, in conjunction with ICCV (2007)
15. Jebara, T., Kondor, R.: Bhattacharyya and expected likelihood kernels. In: Proceedings of Learning theory and Kernel machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24–27, p. 57. Springer, Heidelberg (2003)
16. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 2278–2324 (1998)
17. Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.: Self-taught learning: Transfer learning from unlabeled data. In: Proceedings of the 24th international conference on Machine learning, p. 766. ACM, New York (2007)
18. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
19. Yu, K., Zhang, T., Gong, Y.: Nonlinear Learning using Local Coordinate Coding. In: NIPS (2009)
20. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Supervised dictionary learning. *Adv. NIPS* 21 (2009)
21. Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: Proc. CVPR (2006) (Citeseer)
22. Zhou, X., Cui, N., Li, Z., Liang, F., Huang, T.: Hierarchical Gaussianization for Image Classification. In: ICCV (2009)

# A Discriminative Latent Model of Object Classes and Attributes

Yang Wang and Greg Mori

School of Computing Science, Simon Fraser University, Canada  
{[ywang12,mori](mailto:ywang12,mori)}@cs.sfu.ca

**Abstract.** We present a discriminatively trained model for joint modelling of object class labels (e.g. “person”, “dog”, “chair”, etc.) and their visual attributes (e.g. “has head”, “furry”, “metal”, etc.). We treat attributes of an object as latent variables in our model and capture the correlations among attributes using an undirected graphical model built from training data. The advantage of our model is that it allows us to infer object class labels using the information of both the test image itself and its (latent) attributes. Our model unifies object class prediction and attribute prediction in a principled framework. It is also flexible enough to deal with different performance measurements. Our experimental results provide quantitative evidence that attributes can improve object naming.

## 1 Introduction

What can we say about an object when presented with an image containing it, such as images shown in Fig. 1? First of all, we can represent the objects by their categories, or names (“bird”, “apple”, “chair”, etc.). We can also describe those objects in terms of certain properties or attributes, e.g. “has feather” for (a), “red” for (b), “made of wood” for (c) in Fig. 1.

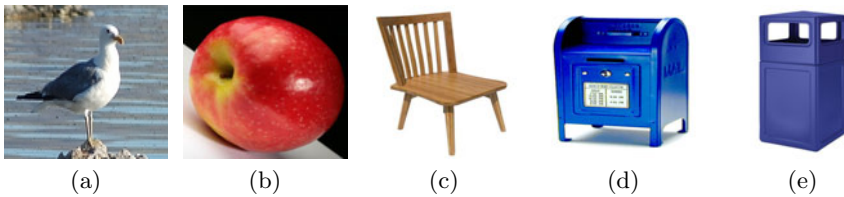
In the computer vision literature, most work in object recognition focuses on the categorization task, also known as *object naming*, e.g. “Does this image window contain a person?” or “Is this an image of a dog (versus cat, chair, table, ...)?”. Some recent work [7,19] proposes to shift the goal of recognition from *naming* to *describing*, i.e. instead of naming the object, try to infer the properties or attributes of objects. Attributes can be parts (e.g. “has ear”), shape (e.g. “is round”), materials (e.g. “made of metal”), color (e.g. “is red”), etc. This attribute-centric approach to object recognition provides many new abilities compared with the traditional naming task, e.g. when faced with an object of a new category, we can still make certain statements (e.g. “red”, “furry”, “has ear”) about it even though we cannot name it.

The concept of attributes can be traced back (at least) to the early work on intrinsic images [1], in which an image is considered as the product of characteristics (in particular, shading and reflectance) of a scene. Conceptually, we can consider shading and reflectance as examples of semantically meaningful



properties (or attributes) of an image. Recently there has been a surge of interest in the computer vision community on learning visual attributes. Ferrari and Zisserman [9] propose a generative model for learning simple color and texture attributes from loose annotations. Farhadi et al. [7] learn a richer set of attributes including parts, shape, materials, etc. Vaquero et al. [18] introduce a video-based visual surveillance system which allows one to search based on people’s fine-grained parts and attributes, e.g. an example could be “show me people with bald head wearing red shirt in the video”.

The attribute-centric approach certainly has great scientific value and practical applications. Some attributes (e.g. “red”) can indeed be recognized without considering object names, and it is possible for people to infer attributes of objects they have never seen before. But object naming is clearly still important and useful. Consider the image in Fig. 1(a), we as humans can easily recognize this object has the attribute “eye”, even though the “eye” corresponds to a very tiny region in the image. Although it is not entirely clear how humans achieve this amazing ability, it is reasonable to believe that we are not running an “eye” detector in our brain in order to infer this attribute. More likely, we infer the object “has eye” in conjunction with recognizing it as a bird (or at least an animal). The issue becomes more obvious when we want to deal with attributes that are less visually apparent. For example, we as humans can recognize the images in Fig. 1(b,c) have the attributes “being edible” and “being able to sit on”, respectively. But those attributes are very difficult to describe in terms of visual appearances of the objects – we infer those attributes most likely because we recognize the objects. In addition, the functions of objects cannot always easily be inferred directly from their visual attributes. Consider the two images in Fig. 1(d,e). They are similar in terms of most of their visual attributes – both are “blue”, “made of metal”, “3D boxy”, etc. But they have completely different functions. Those functions can be easily inferred if we recognize Fig. 1(d) as a mailbox and Fig. 1(e) as a trash can.



**Fig. 1.** Why cannot we forget about object naming and only work on inferring attributes? Look at the image in (a), it is very hard to infer the attribute “has eye” since “eye” is a very tiny region. But we as humans can recognize it “has eyes” most likely because we recognize it is a bird. Other attributes are difficult to infer from visual information alone, e.g. “edible” for (b) and “sit on” for (c). Meanwhile, objects with similar visual attributes, e.g. (d) and (e), can have different functions, which can be easily inferred if we can name the objects.

Our ultimate goal is to build recognition systems that jointly learn object classes and attributes in a single framework. In this paper, we take the first steps toward this goal by trying to answer the following question: can attributes help object naming? Although conceptually the answer seems to be positive, there have only been limited cases supporting it in special scenarios. Kumar et al. [11] show that face verification can benefit from inferring attributes corresponding to visual appearances (gender, race, hair color, etc.) and so-called simile attributes (e.g. a mouth that *looks like* Barack Obama). Attributes have also been shown to be useful in solving certain non-traditional recognition tasks, e.g. when training and test classes are disjoint [7,6,12]. However, when it comes to the traditional object naming task, there is little evidence showing the benefit of inferring attributes. The work in [7] specifically mentions that attribute based representation does not help significantly in the traditional naming task. This is surprising since object classes and attributes are two closely related concepts. Attributes of an object convey a lot of information about the object category, e.g. an object that “has leg” “has head” “furry” should be more likely to be a dog than a car. Similarly, the name of an object also conveys a lot of information about its possible attributes, e.g. a dog tends to “have leg”, and is not likely to “have wing”. The work on joint learning of visual attributes and object classes by Wang and Forsyth [19] is the closest to ours. Their work demonstrates that attribute classifiers and object classifiers can improve the performance of each other. However, we would like to point out that the improvement in their work mainly comes from the fact that the training data are *weakly labeled*, i.e. training data are only labeled with object class labels, but not with exact locations of objects in the image. In this case, an object classifier (say “hat”) and an attribute classifier (say “red”) can help each other by trying to agree on the same location in an image labeled as “red” and “hat”. That work does not answer the question of whether attributes can help object naming without this weakly labeled data assumption, e.g. when an image is represented by a feature vector computed from the whole image, rather than a local patch defined by the location of the object.

Our training data consist of images with ground-truth object class labels (e.g. “person”, “dog”, “chair”, etc.) and attribute labels (e.g. “has torso”, “metal”, “red”, etc.). During testing, we are given a new image without the ground-truth attribute labels, and our goal is to predict the object class label of the test image. We introduce a discriminative model for jointly modelling object classes and attributes. Our model is trained in the latent SVM framework [8]. During testing, we treat the attributes as the latent variables and try to infer the class label of a test image.

The contributions of this paper are three-fold. Firstly and most importantly, we propose a model clearly showing that attributes can help object naming. Our model is also very flexible – it can be easily modified to improve upon many different performance measurements. Secondly, most previous work (e.g. [7,19]) assumes attributes are independent of each other. This is clearly not true. An object that “has ear” is more likely to “has head”, and less likely to be “made of metal”. An important question is how to model the correlations among

attributes. We introduce the *attribute relation graph*, an undirected graphical model built from training data, to capture these correlations. Thirdly, our model can be broadly applied to address a whole class of problems which we call *recognition with auxiliary labels*. Those problems are characterized as classification tasks with certain additional information provided on training data. Many problems in computer vision can be addressed in this framework. For example, in pedestrian detection, auxiliary labels can be the body part locations. In web image classification, auxiliary labels can be the textual information surrounding an image. There has been work that tries to build recognition systems that make use of those auxiliary labels, e.g. [17] for pedestrian detection and [20] for object image classification. However, those work typically use a simple two-stage classification process by first building a system to predict the auxiliary labels, then learning a second system taking into account those auxiliary labels. Conceptually, it is much more appealing to integrate these two stages in a unified framework and learn them jointly, which is exactly what we do in this paper.

## 2 Model Formulation

A training example is represented as a tuple  $(\mathbf{x}, \mathbf{h}, y)$ . Here  $\mathbf{x}$  is the image itself. The object class label of the image is represented by  $y \in \mathcal{Y}$ , where  $\mathcal{Y}$  is a finite label alphabet. The attributes of the image  $\mathbf{x}$  are denoted by a  $K$ -dimensional vector  $\mathbf{h} = (h_1, h_2, \dots, h_K)$ , where  $h_k \in \mathcal{H}_k$  ( $k = 1, 2, \dots, K$ ) indicates the  $k$ -th attribute of the image. We use  $\mathcal{H}_k$  to indicate the set of possible configurations of the  $k$ -th attribute. For example, if the  $k$ -th attribute is “2D boxy”, we will have  $\mathcal{H}_k = \{0, 1\}$ , where  $h_k = 1$  means this object is “2D boxy”, while  $h_k = 0$  means it is not. If the  $k$ -th attribute is “leg”,  $h_k = 1$  means this object “has leg”, while  $h_k = 0$  means it does not. The datasets used in this paper only contain binary-valued attributes, i.e.  $\mathcal{H}_k = \{0, 1\}$  ( $k = 1, 2, \dots, K$ ). For ease of presentation, we will simply write  $\mathcal{H}$  instead of  $\mathcal{H}_k$  from now on when there are no confusions. But we emphasize that our proposed method is not limited to binary-valued attributes and can be generalized to multi-valued or continuous-valued attributes.

We assume there are certain dependencies between some attribute pairs  $(h_j, h_k)$ . For example,  $h_j$  and  $h_k$  might correspond to “head” and “ear”, respectively. Then their values are highly correlated, since an object that “have head” tends to “have ear” as well. We use an undirected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , which we call the *attribute relation graph*, to represent these dependency relations between attribute pairs. A vertex  $j \in \mathcal{V}$  corresponds to the  $j$ -th attribute, and an edge  $(j, k) \in \mathcal{E}$  indicates that attributes  $h_j$  and  $h_k$  have a dependency. We only consider dependencies of pairs of attributes in this paper, but it is also possible to define higher-order dependencies involving more than two attributes. We will describe how to obtain the graph  $\mathcal{G}$  from training data in Sec. 5.

Given a set of  $N$  training examples  $\{(\mathbf{x}^{(n)}, \mathbf{h}^{(n)}, y^{(n)})\}_{n=1}^N$ , our goal is to learn a model that can be used to assign the class label  $y$  to an unseen test image  $\mathbf{x}$ . Note that during testing, we do not know the ground-truth attributes  $\mathbf{h}$  of

the test image  $\mathbf{x}$ . Otherwise the problem will become a standard classification problem and can be solved using any off-the-shelf classification method.

We are interested in learning a discriminative function  $f_{\mathbf{w}} : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$  over an  $\mathbf{x}$  image and its class label  $y$ , where  $\mathbf{w}$  are the parameters of this function. During testing, we can use  $f_{\mathbf{w}}$  to predict the class label  $y^*$  of the input  $\mathbf{x}$  as  $y^* = \arg \max_{y \in \mathcal{Y}} f_{\mathbf{w}}(\mathbf{x}, y)$ . Inspired by the latent SVM [8] (also called the max-margin hidden conditional random field [21]), we assume  $f_{\mathbf{w}}(\mathbf{x}, y)$  takes the following form:  $f_{\mathbf{w}}(\mathbf{x}, y) = \max_{\mathbf{h}} \mathbf{w}^\top \Phi(\mathbf{x}, \mathbf{h}, y)$ , where  $\Phi(\mathbf{x}, \mathbf{h}, y)$  is a feature vector depending on the image  $\mathbf{x}$ , its attributes  $\mathbf{h}$  and its class label  $y$ . We define  $\mathbf{w}^\top \Phi(\mathbf{x}, \mathbf{h}, y)$  as follows:

$$\begin{aligned} \mathbf{w}^\top \Phi(\mathbf{x}, \mathbf{h}, y) = & \mathbf{w}_y^\top \phi(\mathbf{x}) + \sum_{j \in \mathcal{V}} \mathbf{w}_{h_j}^\top \varphi(\mathbf{x}) + \sum_{j \in \mathcal{V}} \mathbf{w}_{y, h_j}^\top \omega(\mathbf{x}) \\ & + \sum_{(j, k) \in \mathcal{E}} \mathbf{w}_{j, k}^\top \psi(h_j, h_k) + \sum_{j \in \mathcal{V}} v_{y, h_j} \end{aligned} \quad (1)$$

The model parameters  $\mathbf{w}$  are simply the concatenation of the parameters in all the factors, i.e.  $\mathbf{w} = \{\mathbf{w}_{h_j}; \mathbf{w}_{y, h_j}; \mathbf{w}_{j, k}; \mathbf{w}_y; v_{y, h_j}\}_{y \in \mathcal{Y}, h_j \in \mathcal{H}, j \in \mathcal{V}, (j, k) \in \mathcal{E}}$ . The details of the potential functions in Eq. (1) are described in the following.

**Object class model  $\mathbf{w}_y^\top \phi(\mathbf{x})$ :** This potential function represents a standard linear model for object recognition without considering attributes. Here  $\phi(\mathbf{x}) \in \mathbb{R}^d$  represents the feature vector extracted from the image  $\mathbf{x}$ , the parameter  $\mathbf{w}_y$  represents a template for object class  $y$ . If we ignore other potential functions in Eq. (1) and only consider the object class model, the parameters  $\{\mathbf{w}_y\}_{y \in \mathcal{Y}}$  can be obtained by training a standard multi-class linear SVM.

In our current implementation, rather than keeping  $\phi(\mathbf{x})$  as a high dimensional vector of image features, we simply represent  $\phi(\mathbf{x})$  as the score of a pre-trained multi-class linear SVM. In other words, we first ignore the attributes in the training data and train a multi-class SVM from  $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^N$ . Then we use  $\phi(\mathbf{x}; y)$  to denote the SVM score of assigning  $\mathbf{x}$  to class  $y$ . Note that we explicitly put  $y$  in the notation of  $\phi(\cdot)$  to emphasize that the value depends on  $y$ . We use  $\phi(\mathbf{x}; y)$  as the feature vector. In this case,  $\mathbf{w}_y$  is a scalar used to re-weight the SVM score corresponding to class  $y$ . This significantly speeds up the learning algorithm with our model. Similar tricks have been used in [3, 22].

**Global attribute model  $\mathbf{w}_{h_j}^\top \varphi(\mathbf{x})$ :** This potential function is a standard linear model trained to predict the label (1 or 0) of the  $j$ -th attribute for the image  $\mathbf{x}$ , without considering its object class or other attributes. The parameter  $\mathbf{w}_{h_j}$  is a template for predicting the  $j$ -th attribute to have label  $h_j$ . If we only consider this potential function, the parameters  $\{\mathbf{w}_{h_j}\}_{h_j \in \mathcal{H}}$  can be obtained via a standard binary linear SVM trained from  $\{(\mathbf{x}^{(n)}, h_j^{(n)})\}_{n=1}^N$ . Similarly, instead of keeping  $\varphi(\mathbf{x})$  as a high dimensional vector of image features, we simply represent it using a scalar  $\varphi(\mathbf{x}; j, h_j)$ , which is the score of predicting the  $j$ -th attribute of  $\mathbf{x}$  to be  $h_j$  by the pre-trained binary SVM.

**Class-specific attribute model  $\mathbf{w}_{y, h_j}^\top \omega(\mathbf{x})$ :** In addition to the global attribute model, we also define a class-specific attribute model for each object class  $y \in \mathcal{Y}$ .

Here  $\mathbf{w}_{y,h_j}$  is a template for the  $j$ -th attribute to take the label  $h_j$  if the object class is  $y$ . If we only consider this potential function,  $\mathbf{w}_{y,h_j}$  ( $h_j \in \{0, 1\}$ ) for a fixed  $y$  can be obtained by learning a binary linear SVM from training examples of object class  $y$ . Similarly, we represent  $\omega(\mathbf{x})$  as a scalar  $\omega(\mathbf{x}; y, j, h_j)$ , which is the score of predicting the  $j$ -th attribute to be  $h_j$  by an SVM pre-trained from examples of class  $y$ .

The motivations for this potential function are two-fold. First, as pointed out by Farhadi et al. [7], learning an attribute classifier across object categories is difficult. For example, it is difficult to learn a classifier to predict the attribute “wheel” on a dataset containing cars, buses, trains. The learning algorithm might end up learning “metallic” since most of the examples of “wheels” are surrounded by “metallic” surfaces. Farhadi et al. [7] propose to address this issue by learning a “wheel” classifier *within a category* and do feature selection. More specifically, they learn a “wheel” classifier from a single object category (e.g. cars). The “wheel” classifier learned in this fashion is less likely to be confused by “metallic”, since both positive and negative examples (i.e. cars with or without “wheel”) in this case have “metallic” attributes. Then they can select features that are useful for differentiating “wheel” from “non-wheel” based on the classifier trained within the car category. The disadvantage of the feature selection approach in [7] is that it is disconnected from the model learning and requires careful manual tuning. Our class-specific attribute model achieves a goal similar to the feature selection strategy in [7], but in a more principled manner since the feature selection is implicitly achieved via the model parameters returned by the learning algorithm.

Second, the same attribute might appear differently across multiple object classes. For example, consider the attribute “leg”. Many object classes (e.g. people, cats) can “have leg”. But the “legs” of people and “legs” of cats can be very different in terms of their visual appearances. If we learn a “leg” attribute classifier by considering examples from both people and cat categories, the learning algorithm might have a hard time figuring out what “legs” look like due to the appearance variations. By separately learning a “leg” classifier for each object category, the learning becomes easier since the positive examples of “legs” within each category are similar to each other. This allows the learning algorithm to use certain visual properties (e.g. furry-like) to learn the “leg” attribute for cats, while use other visual properties (e.g. clothing-like) to learn the “leg” attribute for people.

One might think that the class-specific attribute model eliminates the need for the global attribute model. If this is the case, the learning algorithm will set  $\mathbf{w}_{h_j}$  to be zero. However, in our experiment, both  $\mathbf{w}_{h_j}$  and  $\mathbf{w}_{y,h_j}$  have non-zero entries, indicating these two models are complementary rather than redundant.

**Attribute-attribute interaction  $\mathbf{w}_{j,k}^\top \psi(h_j, h_k)$ :** This potential function represents the dependencies between the  $j$ -th and the  $k$ -th attributes. Here  $\psi(h_j, h_k)$  is a sparse binary vector of length  $|\mathcal{H}| \times |\mathcal{H}|$  (i.e. 4 in our case, since  $|\mathcal{H}| = 2$ ) with a 1 in one of its entries, indicating which of the four possible configurations  $\{(1, 1), (1, 0), (0, 1), (0, 0)\}$  is taken by  $(h_j, h_k)$ , e.g.  $\psi(1, 0) = [0, 1, 0, 0]^\top$ . The parameter  $\mathbf{w}_{j,k}$  is a 4-dimensional vector representing the weights of all those

configurations. For example, if the  $j$ -th and the  $k$ -th attributes correspond to “ear” and “eye”. The entries of  $\mathbf{w}_{j,k}$  that correspond to (1,1) and (0,0) will probably tend to have large values, since “ear” and “eye” tend to appear together in any object.

**Object-attribute interaction**  $v_{y,h_j}$ : This is a scalar indicating how likely the object class being  $y$  and the  $j$ -th attribute being  $h_j$ . For example, let  $y$  correspond to the object class “people” and the  $j$ -th attribute is “torso”, then  $v_{y,1}$  will probably have a large value since most “people” have “torso” (i.e.  $h_j = 1$ ).

### 3 Learning Objective

If the ground-truth attribute labels are available during both training and testing, we can simply consider them as part of the input data and solve a standard classification problem. But things become tricky when we want to take into account the attribute information on the training data, but do not want to “overly trust” this information since we will not have it during testing. In this section, we introduce two possible choices of learning approaches and discuss why we choose a particular one of them.

Recall that an image-label pair  $(\mathbf{x}, y)$  is scored by the function of the form  $f_{\mathbf{w}}(\mathbf{x}, y) = \max_{\mathbf{h}} \mathbf{w}^\top \Phi(\mathbf{x}, \mathbf{h}, y)$ . Given the model parameter  $\mathbf{w}$ , we need to solve the following inference problem during testing:

$$\mathbf{h}^* = \arg \max_{\mathbf{h}} \mathbf{w}^\top \Phi(\mathbf{x}, \mathbf{h}, y) \quad \forall y \in \mathcal{Y} \quad (2)$$

In our current implementation, we assume  $\mathbf{h}$  forms a tree-structured model. In this case, the inference problem in Eq. (2) can be efficiently solved via dynamic programming or linear program relaxation [16,21].

**Learning with latent attributes:** Given a set of  $N$  training examples  $S = \{(\mathbf{x}^{(n)}, \mathbf{h}^{(n)}, y^{(n)})\}_{n=1}^N$ , we would like to train the model parameter  $\mathbf{w}$  that tends to produce the correct label for an image  $\mathbf{x}$ . If the attributes  $\mathbf{h}$  are unobserved during training and are treated as latent variables, a natural way to learn the model parameters is to use the latent SVM [8,21] formulation as follows:

$$\begin{aligned} \min_{\mathbf{w}, \xi} \beta \|\mathbf{w}\|^2 + \sum_{n=1}^N \xi^{(n)} \\ \text{s.t. } \max_{\mathbf{h}} \mathbf{w}^\top \Phi(\mathbf{x}^{(n)}, \mathbf{h}, y^{(n)}) - \max_{\mathbf{h}} \mathbf{w}^\top \Phi(\mathbf{x}^{(n)}, \mathbf{h}, y) \geq \Delta(y, y^{(n)}) - \xi^{(n)}, \forall n, \forall y \end{aligned} \quad (3)$$

where  $\beta$  is the trade-off parameter controlling the amount of regularization, and  $\xi^{(n)}$  is the slack variable for the  $n$ -th training example to handle the case of soft margin,  $\Delta(y, y^{(n)})$  is a loss function indicating the cost of misclassifying  $y^{(n)}$  as  $y$ . In standard multi-class classification problems, we typically use the 0-1 loss  $\Delta_{0/1}$  defined as:

$$\Delta_{0/1}(y, y^{(n)}) = \begin{cases} 1 & \text{if } y \neq y^{(n)} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

**Learning with observed attributes:** Now since we do observe the ground-truth attributes  $\mathbf{h}^{(n)}$  on the training data, one might think a better choice would be to fix those values for  $y^{(n)}$  rather than maximizing over them, as follows:

$$\begin{aligned} \min_{\mathbf{w}, \xi} \beta \|\mathbf{w}\|^2 + \sum_{n=1}^N \xi^{(n)} \\ \text{s.t. } \mathbf{w}^\top \Phi(\mathbf{x}^{(n)}, \mathbf{h}^{(n)}, y^{(n)}) - \max_{\mathbf{h}} \mathbf{w}^\top \Phi(\mathbf{x}^{(n)}, \mathbf{h}, y) \geq \Delta(y, y^{(n)}) - \xi^{(n)}, \forall n, \forall y \end{aligned} \quad (5)$$

The two formulations Eq. (3) and Eq. (5) are related as follows. First, let us define  $\widehat{\mathbf{h}}^{(n)}$  as  $\widehat{\mathbf{h}}^{(n)} = \arg \max_{\mathbf{h}} \mathbf{w}^\top \Phi(\mathbf{x}^{(n)}, \mathbf{h}, y^{(n)})$ . Then it is easy to show that Eq. (3) is a non-convex optimization, while Eq. (5) is convex. In particular, Eq. (5) provides a convex upper-bound on Eq. (3). The bound is tight if  $\widehat{\mathbf{h}}^{(n)}$  and  $\mathbf{h}^{(n)}$  are identical for  $\forall n$ .

**Discussion:** Even though Eq. (5) provides a surrogate of optimizing Eq. (3) as its upper bound, our initial attempt of using the formulation in Eq. (5) suggests that it does not work as well as that in Eq. (3). We believe the reason is the optimization problem in Eq. (5) assumes that we will have access to the ground-truth attributes *during testing*. So the objective being optimized in Eq. (5) does not truthfully mimic the situation at run-time. This will not be an issue if the bound provided by Eq. (5) is tight. Unfortunately, having a tight bound means we need to set the parameters  $\mathbf{w}$  to almost perfectly predict  $\mathbf{h}$  given  $(\mathbf{x}^{(n)}, y^{(n)})$ , which is obviously difficult.

This might be surprising given the fact that the formulation in Eq. (3) seems to ignore some information (i.e. ground-truth attribute labels) during training. At first glance, this argument seems to be reasonable, since Eq. (3) does not require the ground-truth attributes  $\mathbf{h}^{(n)}$  at all. But we would like to argue that this is not the case. The information provided by the ground-truth attributes on training data has been implicitly injected into the feature vectors  $\varphi(\mathbf{x})$  and  $\omega(\mathbf{x})$  defined in the global attribute model and class-specific attribute model (see the descriptions in Sec. 2), since  $\varphi(\mathbf{x})$  and  $\omega(\mathbf{x})$  are vectors of SVM scores. Those scores are obtained from SVM classifiers trained using the ground-truth attribute labels. So implicitly, Eq. (3) already makes use of the information of the ground-truth attributes from the training data. In addition, Eq. (3) effectively models the uncertainty caused by the fact that we do not know the attributes during testing and it is difficult to correctly predict them. So in summary, we choose the **learning with latent attributes** (i.e. non-convex version) formulated in Eq. (3) as our learning objective. But we would like to emphasize that the convex version in Eq. (5) is also a reasonable learning objective. In fact, it has been successfully applied in other applications [3]. We leave the further theoretical and empirical studies of these two different formulations as future work.

## 4 Non-convex Cutting Plane Training

The optimization problem in Eq. (3) can be solved in many different ways. In our implementation, we adopt a non-convex cutting plane method proposed

in [4] due to its ease of use. First, it is easy to shown that Eq. (3) is equivalent to  $\min_{\mathbf{w}} L(\mathbf{w}) = \beta \|\mathbf{w}\|^2 + \sum_{n=1}^N R^n(\mathbf{w})$  where  $R^n(\mathbf{w})$  is a hinge loss function defined as:

$$R^n(\mathbf{w}) = \max_y \left( \Delta(y, y^{(n)}) + \max_{\mathbf{h}} \mathbf{w}^\top \Phi(\mathbf{x}^{(n)}, \mathbf{h}, y) \right) - \max_{\mathbf{h}} \mathbf{w}^\top \Phi(\mathbf{x}^{(n)}, \mathbf{h}, y^{(n)}) \quad (6)$$

The non-convex cutting plane method in [4] aims to iteratively build an increasingly accurate piecewise quadratic approximation of  $L(\mathbf{w})$  based on its sub-gradient  $\partial_{\mathbf{w}} L(\mathbf{w})$ . The key issue here is how to compute the sub-gradient  $\partial_{\mathbf{w}} L(\mathbf{w})$ . Let us define:

$$\begin{aligned} \mathbf{h}_y^{(n)} &= \arg \max_{\mathbf{h}} \mathbf{w}^\top \Phi(\mathbf{x}^{(n)}, \mathbf{h}, y) \quad \forall n, \forall y \in \mathcal{Y} \\ y^{*(n)} &= \arg \max_y \left( \Delta(y, y^{(n)}) + \mathbf{w}^\top \Phi(\mathbf{x}^{(n)}, \mathbf{h}_y^{(n)}, y) \right) \end{aligned} \quad (7)$$

As mentioned in Sec. 2, the inference problem in Eq. (7) can be efficiently solved if the attribute relation graph forms a tree. It is easy to show a sub-gradient  $\partial_{\mathbf{w}} L(\mathbf{w})$  can be calculated as follows:

$$\partial_{\mathbf{w}} L(\mathbf{w}) = 2\beta \cdot \mathbf{w} + \sum_{n=1}^N \Phi(\mathbf{x}^{(n)}, \mathbf{h}_{y^{*(n)}}^{(n)}, y^{*(n)}) - \sum_{n=1}^N \Phi(\mathbf{x}^{(n)}, \mathbf{h}_{y^{(n)}}^{(n)}, y^{(n)}) \quad (8)$$

Given the sub-gradient  $\partial_{\mathbf{w}} L(\mathbf{w})$  computed according to Eq. (8), we can minimize  $L(\mathbf{w})$  using the method in [4]. In order to extend the algorithm to handle more general scenarios involving multi-valued or continuous-valued attributes, we can simply modify the maximization over  $\mathbf{h}$  in Eq. (6,7) accordingly. For example,  $\arg \max_{\mathbf{h}}$  will be replaced by some continuous optimization in the case of continuous attributes.

## 5 Attribute Relation Graph

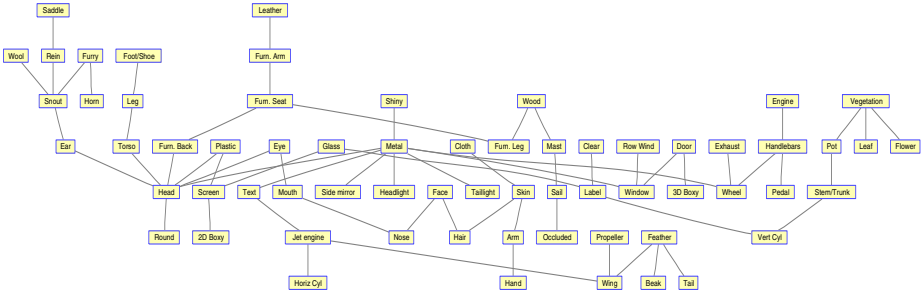
We now describe how to build the attribute relation graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ . In order to keep the inference problem in Eq. (2) tractable, we will assume  $\mathcal{G}$  is a tree-structured graph. Our approach is inspired by the Chow-Liu algorithm [2] for learning Bayesian network structures.

A vertex  $j \in \mathcal{V}$  corresponds to the  $j$ -th attribute. An edge  $(j, k) \in \mathcal{E}$  means the  $j$ -th and the  $k$ -th attributes have dependencies. In practice, the dependencies between certain attribute pairs might be weaker than others, i.e. the value of one attribute does not provide much information about the value of the other one. We can build a graph that only contains edges corresponding to those strong dependencies. The graph  $\mathcal{G}$  could be built manually by human experts. Instead, we adopt an automatic process to build  $\mathcal{G}$  by examining the co-occurrence statistics of attributes in the training data. First, we measure the amount of dependency between the  $j$ -th and the  $k$ -th attributes using the normalized mutual information defined as  $\text{NormMI}(j, k) = \frac{\text{MI}(j, k)}{\min\{H(j), H(k)\}}$ , where  $\text{MI}(j, k)$  is the mutual



information between the  $j$ -th and the  $k$ -th attributes, and  $H(j)$  is the entropy of the  $j$ -th attribute. Both  $MI(j, k)$  and  $H(j)$  can be easily calculated using the empirical distributions  $\hat{p}(h_j)$ ,  $\hat{p}(h_k)$  and  $\hat{p}(h_j, h_k)$  estimated from the training data.

A large  $\text{NormMI}(j, k)$  means a strong interaction between the  $j$ -th and the  $k$ -th attributes. We assign a weight  $\text{NormMI}(j, k)$  to the connection  $(j, k)$ , then run a maximum spanning tree algorithm to find the edges  $\mathcal{E}$  to be included in the attribute relation graph  $\mathcal{G}$ . Similar ideas have been used in [13] to find correlations between video annotations. The attribute relation graph with 64 attributes built from our training data is shown in Fig. 2.



**Fig. 2.** Visualization of the attribute relation graph learned from the training data from the a-Pascal dataset

## 6 Other Loss Functions

This paper mainly deals with multi-class classification problems, where the performance of an algorithm is typically measured by its overall accuracy. It turns out we can modify the learning approach in Sec. 3 to directly optimize other performance measurements. In this section, we show how to adapt the learning objective so it optimizes a more sensible measurement for problems involving highly skewed class distributions.

First we need a new interpretation of Eq. (3). From Eq. (3), it is easy to show  $\xi^{(n)} \geq \Delta(y^*(n), y^{(n)})$ , where  $y^*(n) = \arg \max_y f_{\mathbf{w}}(\mathbf{x}^n, y)$  is the predicted class label of  $\mathbf{x}$  by the model  $f_{\mathbf{w}}$ . So  $\xi^{(n)}$  can be interpreted as an upper bound of the loss incurred on  $\mathbf{x}^{(n)}$  by the model. The cumulative loss on the whole training data is then upper bounded by  $\sum_{n=1}^N \xi^{(n)}$ . In the case of 0-1 loss, the cumulative loss is exactly the number of training examples incorrectly classified by the model, which is directly related to the overall training error. So we can interpret Eq. (3) as minimizing (an upper bound of) the overall training error, with a regularization term  $\beta \|\mathbf{w}\|^2$ .

If the distribution of the classes is highly skewed, say 90% of the data are of a particular class, the overall accuracy is not an appropriate metric for measuring the performance of an algorithm. A better performance measure is the mean per-class accuracy defined as follows. Let  $n_{pq}$  ( $p, q \in \mathcal{Y}$ ) be the number of examples

in class  $p$  being classified as class  $q$ . Define  $m_p = \sum_q n_{pq}$ , i.e.  $m_p$  is the number of examples with class  $p$ . Then the mean per-class accuracy is calculated as  $1/|\mathcal{Y}| \times \left( \sum_{p=1}^{|\mathcal{Y}|} n_{pp}/m_p \right)$ .

We can define the following new loss function that properly adjust the loss according to the distribution of the classes on the training data:

$$\Delta_{\text{new}}(y, y^{(n)}) = \begin{cases} \frac{1}{m_p} & \text{if } y \neq y^{(n)} \text{ and } y^{(n)} = p \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

It is easy to verify that  $\sum_{n=1}^N \Delta_{\text{new}}(y^*(n), y^{(n)})$  directly corresponds to the mean per-class accuracy on the training data. The optimization in Eq. (3) with  $\Delta_{\text{new}}$  will try to directly maximize the mean per-class accuracy, instead of the overall accuracy. This learning algorithm with  $\Delta_{\text{new}}$  is very similar to that with  $\Delta_{0/1}$ . All we need to do is use  $\Delta_{\text{new}}$  in Eq. (3).

Our learning approach can also be extended for detection tasks [8]. In that case, we can adapt our algorithm to directly optimize other metrics more appropriate for detections (e.g. F-measure, area under ROC curve, or the 50% overlapping criterion in Pascal VOC challenge [5]) using the technique in [10,15]. We omit the details due to space constraints. The flexibility of optimizing different performance measurements is an important advantage of the max-margin learning method compared with other alternatives, e.g. the hidden conditional random fields [14].

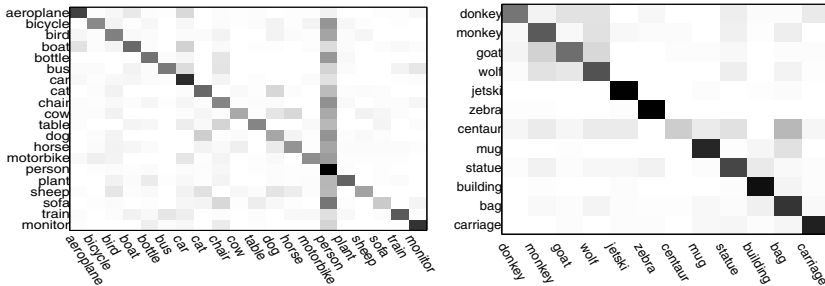
## 7 Experiments

We test our algorithm on two datasets (called **a-Pascal** and **a-Yahoo**) introduced in [7]. The first dataset (a-Pascal) contains 6340 training images and 6355 test images collected from Pascal VOC 2008 challenge. Each image is assigned one of the 20 object class labels: people, bird, cat, cow, dog, horse, sheep, aeroplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, and TV/monitor. Each image also has 64 binary attribute labels, e.g. “2D boxy”, “has hair”, “shiny”, etc. The second dataset (a-Yahoo) is collected for 12 object categories from Yahoo images. Each image in a-Yahoo is described by the same set of 64 attributes. But the object class labels in a-Yahoo are different from those in a-Pascal. Object categories in a-Yahoo are: wolf, zebra, goat, donkey, monkey, statue of people, centaur, bag, building, jet ski, carriage, and mug.

We follow the experiment setup in [7] as close as possible. However, there is one caveat. These two datasets are collected to study the problem of *attribute prediction*, not *object class prediction*. Farhadi et al. [7] use the training images in a-Pascal to learn their model, and test on both the test images in a-Pascal and images in a-Yahoo. We are interested in the problem of *object class prediction*, so we cannot use the model trained on a-Pascal to predict the class labels for images in a-Yahoo, since they have different object categories. Instead, we randomly split a-Yahoo dataset into equal training/testing sets, so we can train a model on a-Yahoo training set and test on a-Yahoo test set.

We use the training images of a-Pascal to build the attribute relation graph using the method in Sec. 5. The graph is shown in Fig. 2. We use the exact same graph in the experiments on the a-Yahoo dataset. In order to do a fair comparison with [7], we use exactly the same image features (called *base feature* in [7]) in their work. Each image is represented as a 9751-dimensional feature vector extracted from information on color, texture, visual words, and edges. Note that since the image features are extracted from the whole image, we have essentially eliminated the weakly labeled data assumption in [19].

Figure 3 (left) shows the confusion matrix of our model trained with  $\Delta_{0/1}$  on the a-Pascal dataset. Table 1 summarizes our results compared with other baseline methods. Since this dataset is heavily biased toward “people” category, we report both overall and mean per class accuracies. Here we show the results of our approach with  $\Delta_{0/1}$  and  $\Delta_{\text{new}}$ . The baseline algorithm is to train an SVM classifier based on the base features. To make a fair comparison, we also report results of SVM with  $\Delta_{0/1}$  and  $\Delta_{\text{new}}$ . We also list the result of the baseline algorithm taken from [7] and the best reported result in [7]. The best reported result in [7] is obtained by performing sophisticated feature selection and extracting more semantic attributes. We can see that both of our models outperform the baseline algorithms. In particular, the mean per class accuracies of our models are significantly better. It is also interesting to notice that models (both our approach and SVMs) trained with  $\Delta_{\text{new}}$  achieve lower overall accuracies than  $\Delta_{0/1}$ , but higher mean per class accuracies. This is exactly what we would expect, since the former optimizes an objective directly tied to the mean per class accuracy, while the latter optimizes one directly tied to the overall accuracy.



**Fig. 3.** Confusion matrices of the classification result of our approach with  $\Delta_{0/1}$  on the a-Pascal (left) and a-Yahoo (right) datasets. Horizontal rows are ground truths, and vertical columns are predictions. Each row is normalized to sum to 1. The mean per class accuracy is calculated by averaging the main diagonal of this matrix. Dark cells correspond to high values.

The results on a-Yahoo are summarized in Table 2. Here we compare with baseline SVM classifiers using the base features. Farhadi et al. [7] did not perform object category prediction on this dataset, so we cannot compare with them. On

this dataset, the performances of using  $\Delta_{0/1}$  and  $\Delta_{\text{new}}$  are relatively similar. We believe it is because this dataset is not heavily biased toward any particular class. So optimizing the overall accuracy is not very different from optimizing the mean per-class accuracy. But the results still show the benefits of attributes for object classification. Figure 3(right) shows the confusion matrix of our approach trained with  $\Delta_{0/1}$  on this dataset.

**Table 1.** Results on the a-Pascal dataset. We report both overall and mean per class accuracies, due to the fact that this dataset is heavily biased toward “people” category

method	overall	mean per-class
Our approach with $\Delta_{0/1}$	<b>62.16</b>	<b>46.25</b>
Our approach with $\Delta_{\text{new}}$	<b>59.15</b>	<b>50.84</b>
SVM with $\Delta_{0/1}$	58.77	38.52
SVM with $\Delta_{\text{new}}$	53.74	44.04
[7] (base features+SVM)	58.5	34.3
[7] (best result)	59.4	37.7

**Table 2.** Results on the a-Yahoo dataset. Similarly, we report both overall and mean per class accuracies

method	overall	mean per-class
Our approach with $\Delta_{0/1}$	<b>78.67</b>	<b>71.45</b>
Our approach with $\Delta_{\text{new}}$	<b>79.88</b>	<b>73.31</b>
SVM with $\Delta_{0/1}$	74.43	65.96
SVM with $\Delta_{\text{new}}$	74.51	66.74

## 8 Conclusion

We have presented a discriminatively trained latent model for joint modelling of object classes and their visual attributes. Different from previous work [7,19], our model encapsulates the correlations among different attributes via the attribute relation graph built from training data and directly optimize the classification accuracy. Our model is also flexible enough to be easily modified according to different performance measurements. Our experimental results clearly demonstrate that object naming can benefit from inferring attributes of objects. Our work also provides a rather general way of solving many other classification tasks involving auxiliary labels. We have successfully applied a similar technique to recognize human actions from still images by considering the human poses as auxiliary labels [22].

## References

1. Barrow, H.G., Tenenbaum, J.M.: Recovering intrinsic scene characteristics from images. In: *Computer Vision Systems*. Academic Press, London (1978)
2. Chow, C.K., Liu, C.N.: Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory* 14(3), 462–467 (1968)
3. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for static human-object interactions. In: *Workshop on Structured Models in Computer Vision* (2010)
4. Do, T.M.T., Artieres, T.: Large margin training for hidden markov models with partially observed states. In: *ICML* (2009)
5. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL visual object classes (VOC) challenge. *IJCV* 88(2), 303–338 (2010)
6. Farhadi, A., Endres, I., Hoiem, D.: Attribute-centric recognition for cross-category generalization. In: *IEEE CVPR* (2010)
7. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: *IEEE CVPR* (2009)
8. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multi-scale, deformable part model. In: *IEEE CVPR* (2008)
9. Ferrari, V., Zisserman, A.: Learning visual attributes. In: *NIPS*. MIT Press, Cambridge (2007)
10. Joachims, T.: A support vector method for multivariate performance measures. In: *ICML* (2005)
11. Kumar, N., Berg, A.C., Belhumeur, P.N., Nayar, S.K.: Attribute and simile classifiers for face verification. In: *IEEE ICCV* (2009)
12. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: *IEEE CVPR* (2009)
13. Qi, G.J., Hua, X.S., Rui, Y., Tang, J., Mei, T., Zhang, H.J.: Correlative multi-label video annotation. In: *ACM Multimedia* (2007)
14. Quattoni, A., Wang, S., Morency, L.P., Collins, M., Darrell, T.: Hidden conditional random fields. *IEEE PAMI* 29(10), 1848–1852 (2007)
15. Ranjbar, M., Mori, G., Wang, Y.: Optimizing complex loss functions in structured prediction. In: *ECCV* (2010)
16. Taskar, B., Lacoste-Julien, S., Jordan, M.I.: Structured prediction, dual extragradient and Bregman projections. *JMLR* 7, 1627–1653 (2006)
17. Tran, D., Forsyth, D.: Configuration estimates improve pedestrian finding. In: *NIPS*. MIT Press, Cambridge (2008)
18. Vaquero, D.A., Feris, R.S., Tran, D., Brown, L., Hampapur, A., Turk, M.: Attribute-based people search in surveillance environments. In: *IEEE Workshop on Applications of Computer Vision* (2009)
19. Wang, G., Forsyth, D.A.: Joint learning of visual attributes, object classes and visual saliency. In: *IEEE ICCV* (2009)
20. Wang, G., Hoiem, D., Forsyth, D.: Building text features for object image classification. In: *IEEE CVPR* (2009)
21. Wang, Y., Mori, G.: Max-margin hidden conditional random fields for human action recognition. In: *IEEE CVPR* (2009)
22. Yang, W., Wang, Y., Mori, G.: Recognizing human actions from still images with latent poses. In: *IEEE CVPR* (2010)

# Seeing People in Social Context: Recognizing People and Social Relationships

Gang Wang<sup>1</sup>, Andrew Gallagher<sup>2</sup>, Jiebo Luo<sup>2</sup>, and David Forsyth<sup>1</sup>

<sup>1</sup> University of Illinois at Urbana-Champaign, Urbana, IL

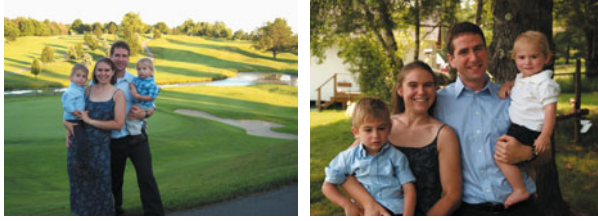
<sup>2</sup> Kodak Research Laboratories, Rochester, NY

**Abstract.** The people in an image are generally not strangers, but instead often share social relationships such as husband-wife, siblings, grandparent-child, father-child, or mother-child. Further, the social relationship between a pair of people influences the relative position and appearance of the people in the image. This paper explores using familial social relationships as context for recognizing people and for recognizing the social relationships between pairs of people. We introduce a model for representing the interaction between social relationship, facial appearance, and identity. We show that the family relationship a pair of people share influences the relative pairwise features between them. The experiments on a set of personal collections show significant improvement in people recognition is achieved by modeling social relationships, even in a weak label setting that is attractive in practical applications. Furthermore, we show the social relationships are effectively recognized in images from a separate test image collection.

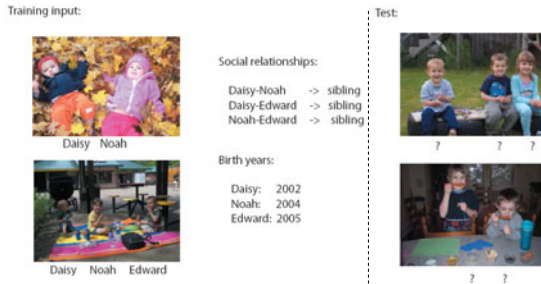
## 1 Introduction

Personal image collections now often contain thousands or tens of thousands of images. Images of people comprise a significant portion of these images. Consumers capture images of the important people in their lives in a variety of social situations. People that are important to the photographer often appear many times throughout the personal collection. Many factors influence the position and pose of each person in the image. We propose that familial social relationships between people, such as “mother-child” or “siblings”, are one of the strong factors. For example, Fig. 1 shows two images of a family at two different events. We observe that the relative position of each family member is the roughly the same. The position of a person relative to another is dependent on both the identity of the persons and the social relationship between them. To explore these ideas, we examine family image collections that have repeating occurrences of the same individuals and the social relationships that we consider are family relationships.

For family image collections, face recognition typically uses features based on facial appearance alone, sometimes including contextual features related to clothing [14, 19, 17]. In essence, that approach makes the implicit assumption that the identity of a face is independent of the position of a face relative to others



**Fig. 1.** Social relationships often exhibit certain visual patterns. For the two people in a wife-husband relationship, the face that is higher in the image is more likely to be the husband. The family members are in roughly the same position in the two images, even though the images are of two different events on different days. The inclination of people to be in specific locations relative to others in a social relationship is exploited in this work for recognizing individuals and social relationships.



**Fig. 2.** In the training procedure, images are weakly labeled. Social relationships and birth years are annotated as input for learning social relationship models. In the recognition test procedure, the goal is to annotate faces present in images with names.

in the image. At its core, our work re-examines this assumption by showing that face recognition is improved by considering contextual features that describe one face relative to others in the image, and that these same features are also related to the familial social relationship.

Our contributions are the following: we develop a probabilistic model for representing the influence between pairwise social relationships, identity, appearance and social context. The experimental results show that adding social relationships results in better performance for face annotation. With the learned relationship models, we can in turn discover social relationships from new image collections where the social relationships are not manually annotated. To the best of our knowledge, this is the first work that shows that explicitly modeling social relationships improves person recognition. Further, this is the first work that demonstrates classification of social relationships from a single image. It is also important to note that our model is learned from an empirically attractive setting of weakly labeled data.

## 1.1 Related Work

Organizing consumer photo collections is a difficult problem. One effective solution is to annotate faces in photos and to search and browse images by people names [12]. Automatic face annotation in personal albums is a hot topic and attracts much attention [3,20]. There has been pioneering work on using social cues for face recognition [6,11,18]. [18] works with strongly labeled data, and only has one type of relationship: friend or not. In comparison, we deal with weakly labeled images, and explicitly model a number of social relationships. In [6], the authors use the social attributes people display in pictures to better recognize genders, ages and identities. However, [6] does not explicitly model different social relationships between people or recognize specific individuals. In [11], recognizing individuals improves by inferring facial attributes. We extend these works by using social relationships as attributes for pairs of people in an image for recognizing people and social relationships.

Weak labeling is an area related to our work. In image annotation, ambiguous labels are related to generic object classes rather than names [18]. Berg et al. [2] is an example where face recognition has been combined with weak labels. In that work, face models are learned from news pictures and captions about celebrities, but ordinary people and the social relationships between them are not considered.

Certainly, the use of social relationships for recognition constitutes a type of context. The social context is related to the social interactions and environment in which an image is captured, and consequently it is not necessarily inferred directly from image data. Our contextual features for describing the relative positions between pairs of people in an image are similar to the contextual features shown to be effective in general object recognition [4,9,15]. In these works, pairwise features enforce priors that, for example, make it unlikely for cows to appear in the sky. We show that our similar features are in fact also useful for improving person recognition and for identifying social relationships. In our work, social relationships act as a high-level context leveraged from human knowledge or human behavior. In this sense, it is similar to the context of [5,16].

## 2 Approach

The common method for providing labeled samples to construct a model of facial appearance for a specific individual involves asking a user to label a set of training faces for each person that is to be recognized. Then, a face model can be learned in a fairly straightforward manner. However, annotating specific faces in a manual fashion is a time-consuming task. In practice, tools such as Flickr, or Adobe Album are used by many consumers, but they only provide weak labels that indicate the presence of a person but not that person's location in the image. Appearance models can still be learned in this scenario, but the label ambiguity increases the learning difficulty. In our work, we assume this realistic weak-labeling scenario, similar to that of [2], and our model is used to disambiguate the labels, learn appearance models, and find the identity of



persons in images that were not in the original training subset. Note also that other frameworks exist for minimizing the effort of the user by using active learning to suggest samples to label [19,10], and our model could be inserted into one of these frameworks.

The procedure is illustrated in Fig. 2. For each image, we only know there are  $N$  names annotated, which are written as  $\{p_i, i = 1, \dots, N\}$ , but do not know the positions or scales of the corresponding faces. Most of faces are automatically detected from images, and we manually add missed faces since we are not studying face detection in this work. Each face is represented by Fisher subspace features. Features of faces are written as  $\{w_j, j = 1, \dots, M\}$ .

We train a face model for each individual. This requires establishing correspondences between names and faces in each training image. Social relationships are manually annotated by photo owners; the relationship between the  $i^{\text{th}}$  and  $j^{\text{th}}$  people is written as  $r_{ij}$ , a discrete variable over the nine pairwise social relationships that we consider. The labeling of this social relationship is reasonable and requires only a small amount of additional effort, because a given pairwise social relationship need be annotated only *once* for the entire personal collection. There are  $N(N - 1)/2$  possible pairwise relationships in one album with  $N$  people, but many pairs of people do not have direct relationships.

**Table 1.** The notation for our model

$p_i$ : the $i^{\text{th}}$ person name	$P$ : all names
$w_i$ : the feature representation of the $i^{\text{th}}$ face	$W$ : all face features
$t_i$ : the age of the $i^{\text{th}}$ person	$T$ : all ages
$r_{ij}$ : the social relationship between the $i^{\text{th}}$ and the $j^{\text{th}}$ person	$R$ : all annotated relationships
$f_{ij}$ : the social relationship features between the $i^{\text{th}}$ and the $j^{\text{th}}$ face	$F$ : all social relationship features
$A$ : the hidden variable which assigns names to faces	$A_i = j$ : the $i^{\text{th}}$ name is assigned to the $j^{\text{th}}$ face
$\theta$ : model parameters	

A specific social relationship usually exhibits common visual patterns in images. For example, in a “husband-wife” relationship, the husband is usually taller than the wife due to physical factors (e.g., the average adult male is 176.8 cm while the average female is 163.3 cm [13]). Of course, it is easy to find exceptions, and this is why our model relies not on “rules” that define the behavior of an individual or a person in a family relationship, but rather on probabilistic distributions of features  $f$  for particular social relationships.

We extract features that reflect social relationships for each pair of faces  $i$  and  $j$ . The features describing the  $i^{\text{th}}$  and  $j^{\text{th}}$  face pair are written as  $f_{ij}$ . This feature vector represents the “social context” in our model. Note that even within a single social relationship, visual patterns are not time-invariant. For example, for “child-mother” relationship, when the child is an infant and the mother is in her 20s, the mother’s face is physically larger than and generally

positioned above the child's; but when the child grows, he or she may eventually have a larger face, be physically taller, and will no longer sit on the mother's lap. To accommodate the evolving roles within a social relationship, we allow the representation of social relationships for different age combinations. This requires that the collection owner provides approximate birth years for each person as illustrated in Fig. 2. In a training image, ages of people are written as  $\{t_i, i = 1, \dots, N\}$ .

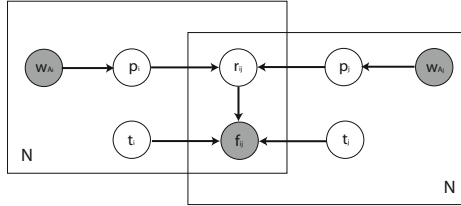


Fig. 3. The graphical model. The notation is explained in Table 1

Given the above defined notations, we then aim to maximize the conditional probability of labels given image observations  $p(P, R, T, W, F)$ , which can be rewritten as:

$$\frac{p(P, R, T, W, F)}{p(W, F)} \sim \sum_A p(P, R, T, W, F | A)p(A) \tag{1}$$

$A$  is a hidden variable that defines the correspondence between faces and names.  $A_i = j$  denotes the  $i^{\text{th}}$  name is assigned to the  $j^{\text{th}}$  face. Given a specific  $A$ , the dependency between  $P, R, T, W$  and  $F$  is represented as shown in Fig. 3. We use a discriminative model to represent the appearance of each name (here we use a weighted KNN classifier due to its robustness, but note that a generative model such as a Gaussian mixture model is also applicable) and generative models for social relationships.

According to the graphical model, (1) can be written as:

$$\sum_A \prod_{i=1}^N p(p_i | w_{A_i}) \prod_{i=1, j=1}^N p(f_{A_i A_j} | r_{ij}, t_i, t_j) p(r_{ij} | p_i, p_j) p(A) \tag{2}$$

where  $w_{A_i}$  denotes the features of the face that is associated with the name  $p_i$ .  $r_{ij}$  is annotated for each pair of names  $p_i$  and  $p_j$ , so  $p(r_{ij} | p_i, p_j)$  is 1 and neglected from now on.  $p(p_i | w_{A_i})$  is calculated as:

$$p(p_i | w_{A_i}) = \frac{\sum_{l=1}^L p(p_i | w_l^{N_{A_i}})}{\sum_{i=1}^N \sum_{l=1}^L p(p_i | w_l^{N_{A_i}})} \tag{3}$$

Where  $w_l^{N_{A_i}}$  denotes the  $l$  nearest neighbor faces found for  $w_{A_i}$  in all the training images.  $p(p_i | w_l^{N_{A_i}}) = 0$  if the image containing  $w_l^{N_{A_i}}$  does not have the person  $p_i$  present.  $\sum_i p(p_i | w_j) = 1$  is enforced in the training procedure.

$f_{A_i A_j}$  denotes the social relationship features extracted from the pair of faces  $A_i$  and  $A_j$ . We extract five types of features to represent social relationships, which are introduced in Section 3. The space of each feature is quantized to several discrete bins, so we can model  $p(f_{A_i A_j}^k | r_{ij}, t_i, t_j)$  as a multinomial distribution, where  $k$  denotes the  $k^{\text{th}}$  type of relationship features. For simplicity, these relationship features are assumed to be independent of each other, and  $p(f_{A_i A_j} | r_{ij}, t_i, t_j)$  could simply be calculated as the product of the probability for each feature. However, we find that the features can be combined in smarter ways. By providing a learned exponent on each probability term, the relative importance of each feature can be adjusted. By learning the exponents with cross-validation on training examples, better performance is achieved.

There are many possible  $t_i$  and  $t_j$  pairwise age combinations, but we may only have a few training examples for each combination. However, visual features do not change much without a dramatic change of age. So we quantize each age  $t_i$  into 5 bins. The quantization partition points are [0 2 17 35 60 100] years. Consequently, there are 25 possible pairwise age bin combinations. For each, we learn a multinomial distribution for each type of relationship feature. The multinomial distribution parameters are smoothed with a Dirichlet prior.

## 2.1 Learning the Model with EM

Learning is performed to find the parameters  $\hat{\theta}$ :

$$\hat{\theta} = \operatorname{argmax}_{\theta} p(P, R, T | W, F; \theta) \quad (4)$$

$\theta$  contains the parameters to define  $p(p | w)$  and  $p(f | r, t)$ . This can not be learned with maximum likelihood estimation because of the hidden variable. Instead, we use the EM algorithm, which iterates between the E step and the M step. Initialization is critical to the EM algorithm. In our implementation, we initialize  $p(p_i | w_j)$  with the parameters produced by the baseline model that omits the social relationship variables. The multinomial distribution is initialized as a uniform distribution.

In the E step, we calculate the probability of the assignment variable  $A$  given the current parameters  $\theta^{\text{old}}$ . For a particular  $A^*$ , we calculate it as:

$$p(A^* | P, R, T, W, F; \theta^{\text{old}}) = \frac{p(P, R, T, W, F | A^*; \theta^{\text{old}})p(A^*; \theta^{\text{old}})}{\sum_A p(P, R, T, W, F | A; \theta^{\text{old}})p(A; \theta^{\text{old}})} \quad (5)$$

$p(P, R, T, W, F, A^*; \theta^{\text{old}})$  can be calculated according to (2). The prior distribution of  $A$  is simply treated as a uniform distribution. This needs to be enumerated over all the possible assignments. When there are a large number of people in images, it becomes intractable. We only assign one  $p_i$  to a  $w_j$  when  $p(p_i | w_j)$  is bigger than a threshold. In this way, we can significantly reduce the number of possible  $A$ .

In the M step, we update the parameters by maximizing the expected likelihood function, which can be obtained by combining (2) and (5). There are two types of parameters, one to characterize  $p(p | w)$  and the other one to characterize  $p(f | r, t)$ . In the M step, when updating one type of parameters using maximum likelihood estimation, the derivative doesn't contain the other type of parameters. Therefore, the updates of parameters for  $p(p | w)$  and  $p(f | r, t)$  are separate. When running the EM algorithm, the likelihood values do not change significantly after 5 to 10 iterations.

## 2.2 Inference

In the inference stage, we are given a test image containing a set of people (without any name label information), we extract their face appearance features  $W$  and relationship features  $F$ , then predict the names  $P$ . We use the relationship models to constrain the labeling procedure, so the classification of faces is not done based on facial appearance alone. This problem is equivalent to finding a one-to-one constraint  $A^*$  in the following way:

$$A^* = \operatorname{argmax}_A p(A | P, R, W, F, T) \quad (6)$$

Here,  $P$  denotes all the names in the dataset. There would be too many possible  $A$  to evaluate and compare. We adopt a simple heuristic by only considering  $A$ s which assign a name  $p$  to a face  $w$  when  $p(p | w)$  is bigger than a threshold. This heuristic works well in our implementation.

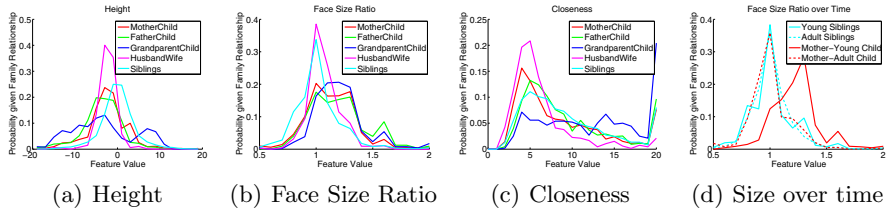
## 3 Implementation Details

In this section, we describe important implementation details. The appearance of each face is represented by projecting the original pixel values into a Fisher subspace learned from a held-out collection (containing no images in common with either the training set or the test set). Each face is represented as a Fisher discriminant space feature.

In our model, the social relationship variable  $r_{ij}$  is discrete over the space of pairwise social relationships. We represent the following nine familial social relationships between a pair of people:

mother-child	father-child	grandparent-child	husband-wife	siblings
child-mother	child-father	child-grandparent	wife-husband	

We consider relationships to be asymmetric (e.g., “mother-child” is different from “child-mother”) because our objective is to identify the role of each person in the relationship. We use the following five types of observed appearance features to represent social relationships.



**Fig. 4.** Pairwise facial features are dependent on social relationships. From these plots, we see that parents’ faces are usually above children’s faces (a), that spouses’ faces are usually about the same size, but are larger than children’s (b), and spouses tend to be close together in an image(c). Note that we also model the changing nature of family relationships over time: a mother’s face is larger than the child’s when the child is young, but they are generally the same size when the child is an adult (d).

**Height:** the height difference is used as a feature. Very simply, we use the ratio of the difference  $y$ -coordinates of the two people’s faces to the average face size of the faces in the image. The ratio is quantized to six bins.

**Face size ratio:** this feature is the ratio of the face sizes. We quantize the ratio to six bins.

**Closeness:** the distance of two people in an image can reveal something about their social relationship. We calculate the Euclidean distance between pair of people, normalized by the average face size. We quantize the distance to five bins.

We train gender and age classifiers based on standard methods, following the examples of [74]. Two linear projections (one for age and one for gender) are learned and nearest neighbors (using Euclidean distance) to the query are found in the projection space.

**Age difference:** we use our age predictor to estimate the ages of people. This age difference, estimated purely from appearance, tells us some information about the social relationship. We quantize age into five ranges, so the age difference between two people has nine possibilities. The age difference relationship is modeled as a multinomial distribution over these nine bins.

**Gender distribution:** the appearance-based gender classifier helps to indicate the role of a person in a social relationship. For example, gender estimates are useful for distinguishing between a wife and husband (or more broadly a heterosexual couple). For each pair of people, there are four possible joint combinations of the genders.

Fig. 4 demonstrates evidence of the dependence between social relationships and our features by showing the distribution of feature values given the social relationships, as learned from our training collections.

## 4 Experiments

In this section, we show experiments that support our assertion that modeling social relationships provides improvements for recognizing people, and allows for the recognition of pairwise social relationships in new images.

In Section 4.1, we examine the task of identifying people through experiments on three personal image collections, each of which has more than 1,000 images and more than 30 distinct people. We show that significant improvement is made by modeling social relationships for face annotation on both datasets. We also investigate how different social relationships features help to boost the performance.

Furthermore, in Section 4.2, we show that learned social relationships models can be transferred across different datasets. Social relationships are learned on a personal image collection, and then social relationships are effectively classified in single images from unrelated separate image collections.

### 4.1 Recognizing People with Social Relationships

In the first experiment, a subset of images from a personal image collection is randomly selected as training examples, and weak name labels are provided for the identities of the people in the images. The remaining images comprise a test set for assessing the accuracy of recognizing individuals. Testing proceeds as follows: First, the correspondence between the names and the faces of the training images are found using the EM procedure from Section 2.1. Next, inference is performed (Section 2.2) to determine the most likely names assignment for each set of faces in each test image. The percentage of correctly annotated faces is used as the measure of performance. This measure is used to evaluate the recognition accuracy in the test set as well as in the training set.

The first collection has 1,125 images and contains 47 distinct people. These people have 2,769 face instances. The second collection contains 1,123 images, with 34 distinct people and 2,935 faces. The third collection has 1,117 images of 152 individuals and 3,282 faces. For each collection, we randomly select 600 images as training examples and the others as test examples. Each image contains at least two people. In total, these images contain 6,533 instances of 276 pairwise social relationships.

**Improvement made by modeling social relationships:** For comparison to our model that includes social relationships, we first perform experiments without modeling social relationships. In the training procedure, we maximize:  $p(P | W) \sim \sum_A \prod_{i=1}^N p(p_i | w_{A_i})p(A)$ . Likewise, the EM algorithm is employed to learn model parameters.

Fig. 5 shows that all datasets show improved recognition accuracy in both training and testing when social relationships are modeled. By modeling social relationships, better correspondence (i.e. disambiguation of the weak label names) in the training set is established. In collection 1, training set accuracy improves by 5.0% by modeling social relationships, and test set identification

improves by 8.6% due to the improved face models as well as the social relationship models. Significant improvement is also observed in collection 2 in both the training (improves by 3.3%) and test (improves by 5.8%) sets. Collection 3 also shows improvement (by 9.5% in training and by 1.8% in testing) although the overall accuracy is lower, mainly because this collection contains many more unique people (152 people versus 47 and 34 in collections 1 and 2).

Fig. 6 illustrates the improvement that modeling social relationships provides for specific test image examples. The faces in green squares are instances that are not correctly classified when the model ignores social relationships, but are corrected by modeling social relationships. We can see that these faces are surrounded by other people who have strong social relationships with, and the visual patterns between people are what is typically expected given their roles in the relationships. The faces in red squares are instances that are correctly classified when appearance alone is considered, but get confused by incorporating social relationships. This is because visual relationship patterns in these pictures are atypical of what is observed in most of other pictures. mother, so she is misclassified as her father, despite her childlike facial appearance.

**Table 2.** Person recognition accuracy in the test set improves for both collections by modeling social relationships using more features. For example, “+height” means that only relative height feature is used, and the other features are omitted.

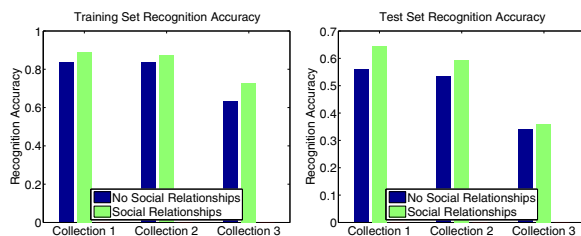
	without relationships	+height	+closeness	+size	+age	+gender	+all
Collection 1	0.560	0.621	0.628	0.637	0.635	0.630	<b>0.646</b>
Collection 2	0.537	0.563	0.560	0.583	0.573	0.584	<b>0.595</b>
Collection 3	0.343	0.361	0.359	<b>0.362</b>	0.362	0.362	0.361
Overall Mean	0.480	0.515	0.516	0.527	0.523	0.525	<b>0.534</b>

**Effect of each social relationship feature:** As described in Section 3, we use five features to encapsulate social relationships. We show how each type of relationship feature helps by in turn omitting all features except that one. The results are shown in Table 2. We observe that relative face size is the most helpful single feature, followed by age and gender. In general, including all features provides significant improvement over using any single feature and adding any single feature is better than using none at all. It is interesting to note that while our results concur with [11] in that we achieve improved face recognition by estimating age and gender.

## 4.2 Recognizing Social Relationships in Novel Image Collections

Our model explicitly reasons about the social relationships between pairs of people in images. As a result, the model has applications for image retrieval based on social relationships.

Social relationships are modeled with visual features such as relative face sizes and age difference, which are not dependent on the identities of people. This



**Fig. 5.** Modeling social relationships improves recognition accuracy. The plots show the improvement in recognition accuracy for both the training set (left) and the test set (right) for two different image collections.

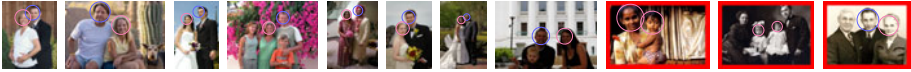


**Fig. 6.** The faces in green squares are instances that are not correctly recognized without modelling social relationships, but are corrected by modeling social relationships. The faces in red squares are correctly recognized at first, but are misrecognized when social relationships are considered. The mistakes are sometimes due to an improbable arrangement of the people in the scene (e.g. the son on the father’s shoulders in the lower right) that is not often observed in the training set. As another example, in the middle image of the second row, the daughter (closer to the camera) appears taller and has a bigger face size than her mother, so she is misclassified as her father, despite her childlike facial appearance.

means social relationship models can be transferred to other image collections with different people. Consequently, the models learned from one image collection can be used to discover social relationships in a separate unrelated image collection with no labeled information at all. We perform two experiments to verify that we learn useful and general models for representing social relationships in images.

In the first experiment, we learn social relationship models from the training examples of collection 1, and classify relationships in collection 2. Because collection 2 contains no “grandparent-child” relationships, we limit the classified  $r_{ij}$  values to the other seven social relationships. The confusion matrix is shown in Fig. 8. Each row of this confusion matrix shows an actual class.





(a) social relationships classified as wife-husband

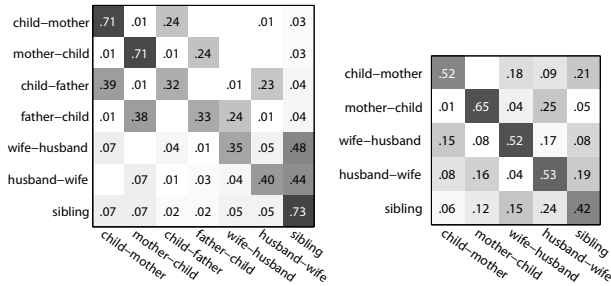


(b) social relationships classified as siblings



(c) social relationships classified as mother-child

**Fig. 7.** Social relationship classification is accomplished from single images with our model, trained only with weak labels on a single, unrelated personal collection. Here, the task is to distinguish between the “wife-husband”, “siblings”, and “mother-child” relationships for each pair of circled faces. Incorrect classifications are outlined in red.



**Fig. 8.** The confusion matrix of social relationships classification. **Left:** We learn social relationship models from collection 1 and test on the images of collection 2. **Right:** We apply the learned social relationship models to a set of images from Flickr, and labeled as one of five social relationships. Both experiments show that social relationship models learned from one collection and transferable and useful for classifying social relationships in images containing strangers.

The averaged value of diagonals is 50.8%, far better than random performance (14.3%). We can see that the mistakes are reasonable. For example, “child-mother” is usually misclassified as “child-father” because the primary visual difference between “mother” and “father” is the gender, which may not be reliably detected from consumer images.

In a second experiment, we perform social relationship recognition experiments on the publicly released group image dataset [6]. First, we manually labeled relationships between pairs of people. A total of 708 social relationships were labeled, at most one relationship per image, and each of the three social relationships has over 200 samples. This dataset is used solely as a test set. The social relationship models are learned from collection 1 in the same weakly supervised learning fashion as before. The confusion matrix is shown in Fig. 8. The overall social relationship classification accuracy in this experiment is 52.7%, again exceeding random classification 20.0%. This performance is significant in that the entire model is trained on a single personal image collection with weak labels. Images classification results from the model are shown for three social relationships in Fig. 7.

## 5 Conclusions

We introduce a model that incorporates pairwise social relationships such as husband-wife or mother-child for representing the relationship between people in a personal image collection. This model is motivated by the observation that the joint appearance between people in an image is associated with both their identities and the social relationship between the pair. We show experimentally several advantages of this representation. First, the model allows for establishing the correspondence between faces and names in weakly labeled images. Second, the identification of unknown faces in test images is significantly improved when social relationship inference is included. Third, social relationships models learned from the weakly labeled data are used to recognize social relationships in single previously unseen images. This work is believed to represent the first attempt at explicitly modeling the pairwise social relationships between people in single consumer images.

**Acknowledgement.** This work was supported in part by the National Science Foundation under IIS -0803603 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. Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D.M., Jordan, M.I.: Matching words and pictures. *JMLR* 3, 1107–1135 (2003)
2. Berg, T., Berg, A., Edwards, J., Maire, M., White, R., Teh, Y.W., Learned-Miller, E., Forsyth, D.: Names and faces in the news. In: *Proc. CVPR* (2004)
3. Chen, L., Hu, B., Zhang, L., Li, M., Zhang, H.J.: Face annotation for family photo album management. *IJIG* 3(1), 81–94 (2003)
4. Divvala, S.K., Hoiem, D., Hays, J., Efros, A., Hebert, M.: An empirical study of context in object detection. In: *Proc. CVPR* (2009)

5. Gallagher, A., Chen, T.: Estimating age, gender, and identity using first name priors. In: Proc. CVPR (2008)
6. Gallagher, A., Chen, T.: Understanding Images of Groups of People. In: Proc. CVPR (2009)
7. Guo, G., Fu, Y., Dyer, C., Huang, T.: Image-based human age estimation by manifold learning and locally adjusted robust regression. In: IEEE Trans. on Image Proc. (2008)
8. Gupta, A., Davis, L.S.: Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 16–29. Springer, Heidelberg (2008)
9. Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. IJCV 80(1), 3–15 (2008)
10. Kapoor, A., Hua, G., Akbarzadeh, A., Baker, S.: Which Faces to Tag: Adding Prior Constraints into Active Learning. In: Proc. ICCV (2009)
11. Kumar, N., Berg, A., Belhumeur, P., Nayar, S.: Attribute and simile classifiers for face verification. In: Proc. ICCV (2009)
12. Naaman, M., Yeh, R., Garcia-Molina, H., Paepcke, A.: Leveraging context to resolve identity in photo albums. In: Proc. JCDL (2005)
13. National Center for Health Statistics. CDC growth charts, United States (2007), <http://www.cdc.gov/nchs/data/nhanes/growthcharts/zscore/stataage.xls>
14. O’Hare, N., Smeaton, A.: Context-aware person identification in personal photo collections. IEEE Trans. MM (2009)
15. Parikh, D., Zitnick, L., Chen, T.: From appearance to context-based recognition: Dense labeling in small images. In: Proc. CVPR (2008)
16. Pellegrini, S., Ess, A., Schindler, K., van Gool, L.: You’ll never walk alone: Modeling social behavior for multi-target tracking. In: Proc. ICCV (2009)
17. Song, Y., Leung, T.: Context-aided human recognition- clustering. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 382–395. Springer, Heidelberg (2006)
18. Stone, Z., Zickler, T., Darrell, T.: Autotagging Facebook: Social network context improves photo annotation. In: Proc. CVPR Internet Vision Workshop (2008)
19. Tian, Y., Liu, W., Xian, R., Wen, F., Tang, X.: A face annotation framework with partial clustering and interactive labeling. In: Proc. CVPR (2007)
20. Zhao, M., Teo, Y.W., Liu, S., Chua, T., Jain, R.: Automatic person annotation of family photo album. In: Sundaram, H., Naphade, M., Smith, J.R., Rui, Y. (eds.) CIVR 2006. LNCS, vol. 4071, pp. 163–172. Springer, Heidelberg (2006)

# Discovering Multipart Appearance Models from Captioned Images

Michael Jamieson, Yulia Eskin, Afsaneh Fazly,  
Suzanne Stevenson, and Sven Dickinson

University of Toronto  
{jamieson, yulia, afsaneh, suzanne, sven}@cs.toronto.edu

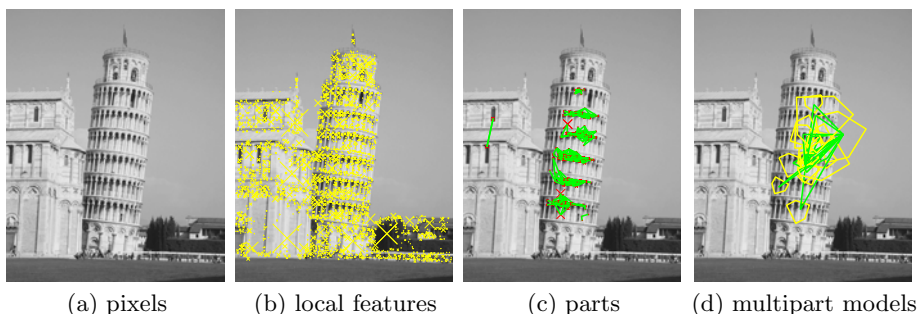
**Abstract.** Even a relatively unstructured captioned image set depicting a variety of objects in cluttered scenes contains strong correlations between caption words and repeated visual structures. We exploit these correlations to discover named objects and learn hierarchical models of their appearance. Revising and extending a previous technique for finding small, distinctive configurations of local features, our method assembles these co-occurring parts into graphs with greater spatial extent and flexibility. The resulting multipart appearance models remain scale, translation and rotation invariant, but are more reliable detectors and provide better localization. We demonstrate improved annotation precision and recall on datasets to which the non-hierarchical technique was previously applied and show extended spatial coverage of detected objects.

## 1 Introduction

Computer vision tasks from image retrieval to object class recognition are based on discovering similarities between images. For all but the simplest tasks, meaningful similarity does not exist at the level of basic pixels, and so system designers create image representations that abstract away irrelevant information. One popular strategy for creating more useful representations is to learn a hierarchy of parts in which parts at one level represent meaningful configurations of sub-parts at the next level down. Thus salient patterns of pixels are represented by local features, and recurring configurations of features can, in turn, be grouped into higher-level parts, and so on, until ideally the parts represent the objects that compose the scene. The hierarchical representations are inspired by and intended to reflect the compositional appearance of natural objects and artifacts. For instance, each level of the Leaning Tower of Pisa appears as a ring of arches while the tower as a whole is composed of a (nearly) vertical stack of levels.

With this strategy in mind, we build upon the approach of [1] to produce a system with more accurate image annotation and improved object localization. Given images of cluttered scenes, each associated with potentially noisy captions, our previous method [1] can discover configurations of local features that strongly correspond to particular caption words. Our system improves the overall

distribution of these local configurations to optimize the overall correspondence with the word. While individual learned parts are often sufficient to indicate the presence of particular exemplar objects, they have limited spatial extent and it is difficult to know whether a collection of part detections in a particular image are from multiple objects or multiple parts of a single object. Our system learns meaningful configurations of parts wherever possible, allowing us to reduce false annotations due to weak part detections and provide a better indication of the extent of detected objects. Figure 1 illustrates how low-level features are assembled in stages to form a multipart model (MPM) for the Leaning Tower. MPMs are more robust to occlusion, articulation and changes in perspective than a flat configuration of features. While the instantiated system uses exemplar-specific SIFT features, the framework can support more categorical features.



**Fig. 1.** Object model detection and learning progresses in stages. Gradient patterns in the original image (a) are grouped into local features (b). Configurations of local features with strong word correspondence are captured as part models (c). Finally, we represent meaningful configurations of part models as multipart models (d).

## 2 Related Work

A number of researchers have studied the problem of automatic image annotation in recent years [2,3,4,5,6,1]. Given cluttered images of multiple objects paired with noisy captions, these systems can learn meaningful correspondences between caption words and appearance models.

In many automatic annotation systems, the main component of the appearance model is a distribution over colors and textures. This kind of representation is a good fit for relatively structureless materials such as grass, sand or water and is relatively robust to grouping or segmentation errors. However, objects such as buildings and bicycles often lack a distinctive color or texture, and require representations that can capture a particular configuration of individually ambiguous parts. Most of these automatic annotation systems do not focus on learning such feature configurations. Often, appearance is modeled as a mixture of features (*e.g.*, [5,3,6]) in which common part configurations are reflected in

co-occurrence statistics but without spatial information. Similarly, the Markov random field model proposed by Carbonetto *et al.* [4] can represent adjacency relationships but not spatial configurations.

In contrast, the broader object recognition literature contains many methods for grouping individual features into meaningful configurations and even arranging features into hierarchies of parts. For instance, Fergus *et al.* [7] and Crandall and Huttenlocher [8] look for features and relationships that recur across a collection of object images in order to learn object appearance models consisting of a distinctive subset of features and their relative positions. A natural strategy to improve the flexibility and robustness of such models is to organize the object representation as a parts hierarchy (*e.g.*, [9,10,11,12,13,14]). The part hierarchy can be formed by composing low-level features into higher and higher level parts (*e.g.* Kokkinos and Yuille [9], Zhu *et al.* [10]) or by decomposing larger-scale shared structures into recurring parts (*e.g.*, Epshtein and Ullman [13]). The composition and learning method of parts at different levels of the hierarchy may be highly similar (*e.g.*, Bouchard and Triggs [11], Fidler *et al.* [12]) or heterogeneous (*e.g.*, Ommer and Buhmann [14]). Some of these methods can learn an appearance model from training images with cluttered backgrounds, sometimes without relying on bounding boxes. However, unlike most automatic annotation work, they are not designed for images containing multiple objects and multiple annotation words.

In [1], we describe an automatic annotation system that can capture explicit spatial configurations of features while retaining the ability to learn from noisy, unstructured collections of captioned images. Guided by correspondence with caption words, the system iteratively constructs appearance graphs in which vertices represent local features and edges represent spatial relationships between them. However, the learned appearance models usually have limited spatial extent, with each model typically describing only a distinctive portion of an object. There is no way to determine whether a set of detections in a given image represents multiple objects or different parts of the same object. Our system addresses these limitations by using the appearance models as parts in larger hierarchical object models.

### 3 Images, Parts and Multipart Models

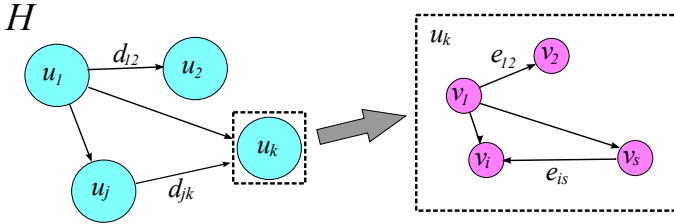
Our system learns multipart appearance models (MPMs) by detecting recurring configurations of lower-level ‘parts’ that together appear to have a strong correspondence with a particular caption word. Though our overall approach could be appropriate for a variety of part features, in this paper our parts are local configurations of interest points as in [1].

In [1], an image is represented as a set of local interest points,  $I = \{p_m | m = 1 \dots |I|\}$ . These points are detected using Lowe’s SIFT method [15], which defines each point’s spatial coordinates,  $\mathbf{x}_m$ , scale  $\lambda_m$  and orientation  $\theta_m$ . A PCA-SIFT [16] feature vector ( $\mathbf{f}_m$ ) describes the portion of the image around each point. In addition, a vector of transformation-invariant spatial relationships  $r_{mn}$  is defined

between each pair of points,  $p_m$  and  $p_n$ , including the relative distance between the two points ( $\Delta x_{mn}$ ), the relative scale difference between them ( $\Delta \lambda_{mn}$ ) and the relative bearings in each direction ( $\Delta \phi_{mn}$ ,  $\Delta \phi_{nm}$ ).

A part appearance model describes the distinctive appearance of an object part as a graph  $G = (V, E)$ . Each vertex  $v_i \in V$  is composed of a continuous feature vector  $\mathbf{f}_i$  and each edge  $e_{ij} \in E$  encodes the expected spatial relationship between two vertices,  $v_i$  and  $v_j$ . Model detections have a confidence score,  $\text{Conf}_{\text{detect}}(O, G) \in [0, 1]$ , based on the relative likelihood of an observed set of points  $O$  and the associated spatial relations being generated by the part model  $G$  versus unstructured background.

Multipart models are very similar in structure to the local appearance models described in [1]. As shown in Figure 2, a multipart model is a graph  $H = (U, D)$  where vertices  $u_j, u_k \in U$  are part appearance model detections and each edge  $d_{jk} \in D$  encodes the spatial relationships between them, using the same relationships as in the part model:  $d_{jk} = (\Delta x_{jk}, \Delta \lambda_{jk}, \Delta \phi_{jk}, \Delta \phi_{kj})$ .



**Fig. 2.** A multipart model  $H$  is a graph with parts  $u_j \in U$  and spatial relationships  $d_{jk} \in D$ , where each part is a graph  $G$  with local features  $v_i \in V$  and spatial relationships  $e_{is} \in E$

## 4 Discovering Parts

Multipart models are composed of the same type of individual appearance models that were discovered in [1]. However, models trained to maximize stand-alone detection performance are generally not ideal as parts of a larger appearance model. Singleton appearance models need to act as high-precision detectors while MPM parts can be individually more ambiguous and rely on the MPM layer to weed out false-positive detections by imposing co-occurrence and spatial constraints. Therefore, when learning MPM parts, we can accept some loss of precision in exchange for better recall and better spatial coverage of the object of interest. We implement this shift toward weaker parts with better coverage by replacing the part initialization process in [1] with our own improved process and by limiting the size of learned part models to eight vertices.

#### 4.1 Model Initialization through Image Pair Sampling

We replace the clustering-based model initialization method of [1] with an approach that makes earlier use of language information. The system in [1] summarizes the visual information within each neighborhood of an image set as a quantized bag-of-features descriptor called a *neighborhood pattern* and then uses clustering to group similar neighborhood patterns. Next, the system checks for promising correspondences between the occurrence patterns of each neighborhood cluster and each word. Finally, clusters with the best correspondences for each word are used to extract initial two-vertex appearance models.

This clustering approach has several drawbacks. The neighborhood patterns are noisy due to features quantization and detector errors. Therefore a low similarity threshold is needed to reliably group similar appearances. However, this allows unrelated neighborhoods to join the cluster. Especially on large image sets, this can add substantial noise to the cluster occurrence pattern, obscuring its true word correspondences. Therefore recurring visual structure corresponding to rarer object views is often overlooked.

Our initialization method avoids feature quantization and uses word labels early-on in the process. Instead of using a neighborhood pattern, we compare visual features directly. Rather than cluster visual structure across the entire training set, we look for instances of shared appearance between pairs of images with the same word label. For a given word  $w$ , the system randomly samples pairs of images  $I_A$  and  $I_B$  from those with captions containing  $w$  and identifies neighborhoods in the two images that share visual structure.

We identify shared neighborhoods in three steps. First, the system looks for uniquely-matching features that are potential anchors for shared neighborhoods. Following [15], we identify matching features that are significantly closer to each other than to either feature’s second-best match, *i.e.*, features  $\mathbf{f}_m \in I_A$  and  $\mathbf{f}_n \in I_B$  that satisfy equations [1] and [2]

$$|\mathbf{f}_m - \mathbf{f}_n|^2 \leq \psi_u |\mathbf{f}_m - \mathbf{f}_k|^2, \forall \mathbf{f}_k \in \{I_B - \mathbf{f}_n\} \quad (1)$$

$$|\mathbf{f}_m - \mathbf{f}_n|^2 \leq \psi_u |\mathbf{f}_l - \mathbf{f}_n|^2, \forall \mathbf{f}_l \in \{I_A - \mathbf{f}_m\} \quad (2)$$

where  $\psi_u < 1$  controls degree of uniqueness of anchor matches. For each pair of uniquely-matching features, the system checks for supporting matches in the surrounding neighborhood. These supporting matches aren’t required to be unique, so the corresponding uniqueness quantifier  $\psi_s > 1$ . For each supporting match pair  $f_i \in I_A$  and  $f_j \in I_B$ , the system then verifies that the spatial relationships between the unique feature and the supporting feature in the two images ( $r_{mi}$  and  $r_{nj}$ ) are consistent. A shared neighborhood has a pair of unique matches and at least two spatially consistent supporting matches.

Given this evidence of shared visual structure, we construct a set of two-vertex part models, each with one vertex based on the unique match and the other on a strong supporting match. These two-vertex models represent shared visual structure between two images labeled with word  $w$ . To check whether the models correspond with  $w$ , the system detects each model  $G$  across the training image



set and compares its occurrence pattern with that of  $w$ . Below, we explain how we sample image pairs and filter the resulting initial part models to maximize overall coverage of the object.

## 4.2 Part Coverage Objective

In [1], the system develops the  $n$  neighborhood clusters with the best correspondence with  $w$  into full appearance models. This approach concentrates parts on the most common views of an object, neglecting less common views and appearances associated with  $w$ . Our method instead selects initial part models so that, *as a group*, they have good coverage of  $w$  throughout the training set.

Ideally, a set of part models  $\mathcal{G}$  would have multiple, non-overlapping detections in every training set image annotated with word  $w$  and no detections elsewhere. We represent the distribution of model detections throughout the  $k$  training images with the vector  $\mathbf{Q}_w = \{Q_{wi} | i = 1, \dots, k\}$ . If  $n_i$  is the number of independent model detections in image  $i$ ,  $Q_{wi} = 1 - \nu^{n_i}$ ,  $\nu < 1$ . With multiple detections,  $Q_{wi}$  approaches 1, but each successive detection has a smaller effect.

We evaluate how well  $\mathcal{G}$  approximates the ideal by evaluating the correspondence between  $\mathbf{Q}_w$  and a vector  $\mathbf{r}_w$  indicating images with  $w$  in the caption using an F-Measure,  $F(\mathbf{r}_w, \mathbf{Q}_w)$ . The part initialization process greedily grows and modifies a collection of non-overlapping two-vertex part models  $\mathcal{G}$  to maximize  $F(\mathbf{r}_w, \mathbf{Q}_w)$ . At each iteration, it draws a pair of images from the sample distribution  $\mathbf{s}_w$  and uses them to generate potential part models.  $\mathbf{Q}_w$  influences the sample distribution:  $\mathbf{s}_w \sim 1 - \mathbf{Q}_w$ . This focuses the search for new models in images that do not already contain several model detections. The algorithm calculates, for each potential model, the effects on the correspondence score  $F$  of adding the model to the current part set, of replacing each of the models in the current set and of rejecting the model. The algorithm implements the option which leads to the greatest improvement in correspondence. The process stops once no new models have been accepted in the last  $N_{pairs}$  image-pair samples.

Besides optimizing the explicit objective function, the initialization system also avoids redundant models with many overlapping detections. Two models are considered to be redundant when their detections overlap nearly as often as they occur separately. When a new two-vertex model is considered, if selected it must replace any models that it makes redundant.

## 5 Building Multipart Models

After learning distinctive part models, but before assembling them into multipart models, we perform several stages of processing. Algorithm 1 summarizes both the preprocessing steps and the MPM initialization and assembly process, with reference to the subsections below that explain the steps of the algorithm.

---

**Algorithm 1.** Uses parts associated with word  $w$  to assemble multipart models.

---

ConstructMPMs( $w$ )

1. For each part  $G$  associated with  $w$ , find the set  $\mathcal{O}_G$  of observations of  $G$  in training images.
  2. Identify and remove redundant parts (section 5.1).
  3. For each  $G$ , set the spatial coordinates of each observation  $O_G \in \mathcal{O}_G$  (section 5.2):
    - Choose representative vertex  $v_c$  to act as center of  $G$ .
    - For each  $v_i \in \mathbf{v}_G$ , find average relationship,  $\bar{\mathbf{r}}_{ic}$ , between co-occurences of  $(v_i, v_c) \in \mathcal{O}_G$ .
    - For each  $O_G \in \mathcal{O}_G$ , and each observed vertex  $\mathbf{p}_i \in O_G$  calculate expected position of  $\mathbf{x}_c$  based on  $(\bar{\mathbf{r}}_{ic}, \mathbf{x}_i)$ . Part spatial coordinate  $\mathbf{x}_G$  is the average expected center  $\bar{\mathbf{x}}_c$ .
  4. Sort parts by  $\text{Conf}_{corr}(G, w)$ .
  5. For each  $G$ :
    - Skip expansion if most  $O_G \in \mathcal{O}_G$  are already incorporated into existing MPMs (section 5.3).
    - Iteratively expand  $G$  into an MPM  $H$  using same method as part models (section 5.4):
      - Expand MPM  $H$  to  $H^*$  by adding new part or spatial relationship.
      - Detect  $H^*$  across the training image set (section 5.5).
      - If new MPM–word correspondence,  $\text{Conf}_{corr}(H^*, w) > \text{Conf}_{corr}(H, w)$ ,  $H \leftarrow H^*$ .
    - If at least  $N_{MPM}$  multipart models have been created, return.
- 

## 5.1 Detecting Duplicate Parts

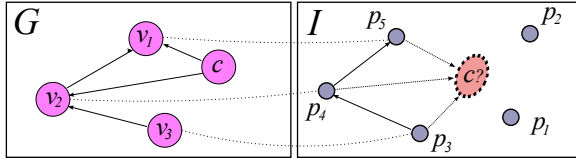
Our initialization method avoids excessive overlap of initial part models. However, during model refinement, two distinct part models can converge to cover the same portion of an object’s appearance. Near-duplicate parts must be pruned or they could complicate the search for multipart models since they could be interpreted as a pair of strongly co-occurring, independent parts.

Rather than detect near-duplicates by searching for partial isomorphisms between part models, we look for groups of parts that tend to be detected in the same images at overlapping locations. If a vertex  $v_{A_i}$  in model  $G_A$  maps to the same image point as vertex  $v_{B_j}$  in model  $G_B$  in more than half of detections, then we draw an equivalence between  $v_{A_i}$  and  $v_{B_j}$ . If more than half of the vertices in either part are equivalent, we remove the part with the weakest word–model correspondence confidence  $\text{Conf}_{corr}(G, w)$ .

## 5.2 Locating Part Detections

The parts described in [1] encode spatial relationships among local interest points; we construct multipart models by discovering spatial relationships between such detected parts. However, while a local interest point detector provides that point’s scale, orientation and location, the part detector does not. We therefore set the spatial coordinates for each part detection based on the underlying image points in a way that is robust to occlusion and errors in feature detection.

For each part we select a central vertex and for each detection we estimate the center’s spatial coordinates. The center vertex need not be observed in every detection, since each observed vertex contributes to a weighted estimate of the center’s coordinates. Figure 3 illustrates this approach. We use the estimated location and orientation of the center and multiply the estimated scale of the center vertex by a part-specific factor so that the detected part scale reflects the normal spread of the part’s vertices.



**Fig. 3.** The spatial coordinates of a part detection are tied to a central vertex  $c$ . We estimate  $c$ 's coordinates based on observed vertices, even if  $c$  itself is not observed.

### 5.3 Choosing Initial Multipart Models

Our system uses the most promising individual part models as seeds for constructing multipart models. Parts that have good correspondence with a word are likely to co-occur with other parts in stable patterns from which large MPMs with good spatial coverage can be constructed. However, if only the strongest part models are expanded, the resulting MPMs may be too clustered around only the most popular views of the object. This would neglect views with weaker individual parts where MPMs can make the biggest difference in precision.

Therefore initial model selection proceeds as follows. Part models are evaluated in the order of their correspondence with a word  $w$ . A model is expanded if at least half of its ‘good’ detections (in images labeled with  $w$ ) have not been incorporated into any of the already-expanded MPMs. Selective expansion continues until the list of part models is exhausted or  $N_{MPM}$  distinct multipart models have been trained for a given word.

### 5.4 Refinement and Expansion of Multipart Models

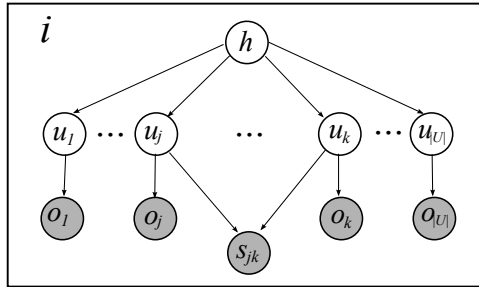
In order to expand the multipart models, we take an approach very similar to [1], in that we use the correspondence strength  $\text{Conf}_{corr}(H, w)$  between a multipart model  $H$  and word  $w$  to guide the expansion of these two-vertex graphs into larger multipart models. Introduced in [1], the correspondence score reflects the amount of evidence, available in a set of training images, that a word and a part model are generated from a common underlying source object, as opposed to appearing independently.

Each iteration of the expansion algorithm begins by detecting all instances of the current multipart model in the training set (section 5.5) and identifying additional parts that tend to co-occur with a particular spatial relationship relative to the multipart model. We propose an expansion of the MPM  $H$  either by adding a new part model and spatial relationship from among these candidates or by adding a new edge between existing vertices. The proposal is accepted if it improves  $\text{Conf}_{corr}(H, w)$  (starting a new iteration), and rejected otherwise. The expansion process continues until potential additions to  $H$  have been exhausted.

### 5.5 Detecting Multipart Models

As in part model detection, multipart detection must be robust to changes in viewpoint, occlusion and lighting that can cause individual part detections to be somewhat out of place or missing entirely. We use a simple generative model illustrated in Figure 4 to explain the pattern of part detections both in images that contain a particular multipart model and those that do not.

Each image  $i$  has an independent probability  $P(h_i = 1)$  of containing the multipart model  $H$ . Given  $h_i$ , the presence of each model part is determined independently ( $P(u_{ij} = 1|h_i)$ ). The foreground probability of a model part being present is relatively high ( $P(u_{ij} = 1|h_i = 1) = 0.95$ ), while the background probability,  $P(u_{ij} = 1|h_i = 0)$ , is equal to its normalized frequency across the training image set. If a part is present, it tends to have a higher observed detection confidence,  $o_{ij}$  ( $p(o_{ij}|u_{ij} = 1) = 2o_{ij}$ ,  $p(o_{ij}|u_{ij} = 0) = 2(1 - o_{ij})$ ). If the multipart model is present ( $h_i = 1$ ) and contains an edge  $r_{jk}$ , and the parts  $u_{ij}$  and  $u_{ik}$  are present, then the observed spatial relationship  $s_{ijk}$  between the two parts has a relatively narrow distribution centered at the edge parameters. Otherwise, all spatial relationships follow a broad background distribution.



**Fig. 4.** A graphical model of the generative process with multipart model indicator  $h$ , part indicators  $\mathbf{u}$ , part detection confidences  $\mathbf{o}$  and observed spatial relations  $\mathbf{s}$

In any given image, there may be many possible assignments between multipart model vertices and observed part detections. We choose assignments in a greedy fashion in order to maximize  $P(h_i = 1|\mathbf{o}_i, \mathbf{s}_i)$ . First we choose the best-fit assignment of two linked vertices, then one by one we choose the vertex assignment that makes the largest improvement in  $P(h_i = 1|\mathbf{o}_i, \mathbf{s}_i)$  and is consistent with existing assignments.

The prior probability  $P(h_i = 1)$  depends on the complexity of the MPM, with more complex multipart models having a lower prior probability. Specifically:

$$P(h_i = 1) = \alpha^{|\mathbf{U}|} \cdot \beta^{|\mathbf{D}|}. \tag{3}$$

where  $\alpha, \beta < 1$  and  $|\mathbf{U}|$  and  $|\mathbf{D}|$  are, respectively, the number of vertices and edges in  $H$ . The constants  $\alpha$  and  $\beta$  were selected based on detection experiments on

random synthetic MPMs with a wide range of sizes in order to prevent large, complex models from being detected when only a tiny fraction of their vertices are present.

## 6 Results

Once we have discovered a set of individual part models and learned multipart models from configurations of the parts, we can use these learned structures to annotate new images. We begin by detecting all part models in the image (even those that are relatively weakly detected or have relatively low individual correspondence confidence). Based on these part observations, we then evaluate detection confidence for all learned MPMs. Following [1], our annotation confidence for both parts and multipart models is the product of detection confidence,  $\text{Conf}_{\text{detect}}(i, H)$ , and correspondence confidence  $\text{Conf}_{\text{corr}}(H, w)$ . Overall annotation confidence is the maximum annotation confidence over word  $w$ 's detected models in image  $i$ .

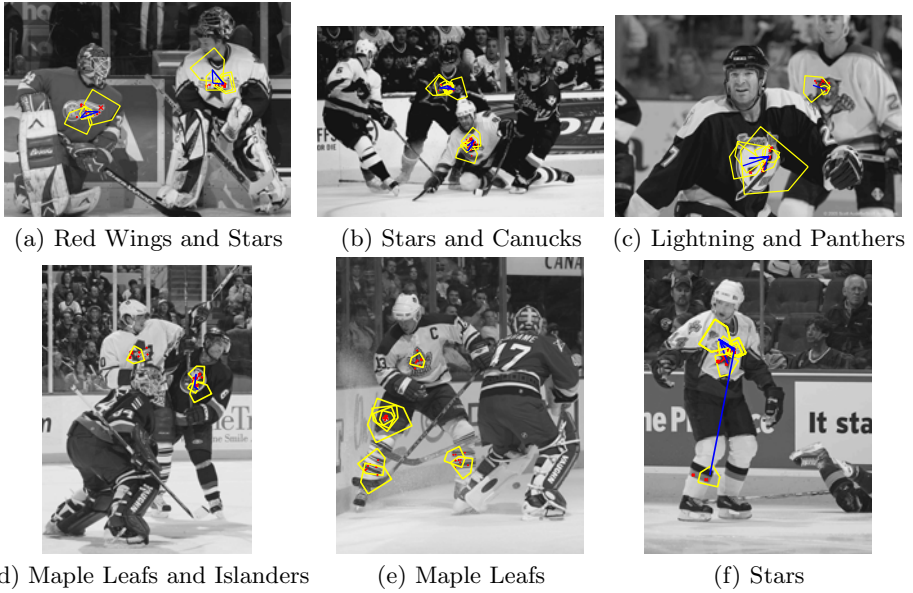
For ease of comparison, we ran our system on three image sets described in [1]. In all three cases, the changes to part initialization combined with the addition of MPM models improve the precision and recall of annotation on new images compared to the system in [1]. The degree of improvement seems to depend on the scale and degree of articulation of named objects.

In experimentation on the small TOYS image set, we find that the particular values of our system parameters do not have a significant effect on our results. The same parameter values chosen based on the TOYS set results are carried over to the two larger and more significant sets without modification. We set uniqueness factors  $\psi_u = 0.9$  and  $\psi_s = 1.2$ .  $N_{\text{pairs}} = 50$  allows a large number of failed pair samples before ending initial model search.  $\nu = 0.75$  allows  $Q_{wi}$  to build gradually. We set the maximum number of MPMs per word,  $N_{\text{MPM}} = 25$ , more than the number of distinct views available for individual objects in these image collections. Finally, we choose MPM detection parameters  $\alpha = 0.25$  and  $\beta = 0.33$  based on experiments on synthetic data.

The first set, TOYS, is a small collection of 228 images of arrangements of children's toys captured and annotated by the authors of [1]. For the sake of completeness, we report our results on this set while focusing on the larger and more natural HOCKEY and LANDMARK sets. Without MPMs, our new model initialization method modestly improves recall on the TOYS set while slightly lowering overall precision. Including MPMs corrects precision, resulting in a net improvement in recall of about 3% at 95% precision.

### 6.1 Experiments on the HOCKEY Data Set

The HOCKEY set includes 2526 images of National Hockey League (NHL) players and games, with associated captions, downloaded from a variety of sports websites. It contains examples of all 30 NHL teams and is divided into 2026 training and 500 test image-caption pairs. About two-thirds of the captions are

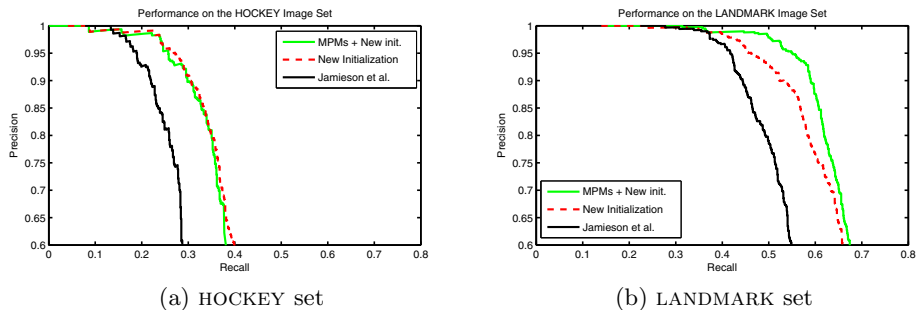


**Fig. 5.** Sample detections of objects in the HOCKEY test set. Part detections are drawn in yellow, supporting interest points in red and spatial relationships in blue.

full sentence descriptions, whereas the remainder simply name the two teams involved in the game.

Figure 5 shows sample multipart model detections on test-set images and the associated team names. Compared to MPMs in the TOY and LANDMARK sets, most MPMs in the HOCKEY set are relatively simple. They typically consist of 2 to 4 parts clustered around the team’s chest logo. Since the chest logos are already reasonably well covered by individual part models, there is little reward for developing extensive MPMs. In principle, MPMs could tie together parts that describe other sections of the uniform (socks, pants, shoulder insignia) like those shown in Figure 5(e), but this type of MPM (seen in Figure 5(f)) is quite rare. There may be too much articulation and too few instances of co-occurrence of these parts in the training set to support such MPMs.

Figure 6(a) indicates that our new approach for initializing part models leads to about a 12% improvement in recall. Considering the barriers to achieving high recall on the HOCKEY set (discussed in 11), this represents a substantial gain. Our initialization system is better able to identify regions of distinctive appearance than the approach in 11. For instance, one of the best-recognized NHL teams using our method was completely undetected in 11. On the other hand, the addition of MPMs does not improve annotation performance at all. This is probably due to the relatively small size of distinctive regions in the HOCKEY images combined with a degree of articulation and occlusion that make larger models unreliable.



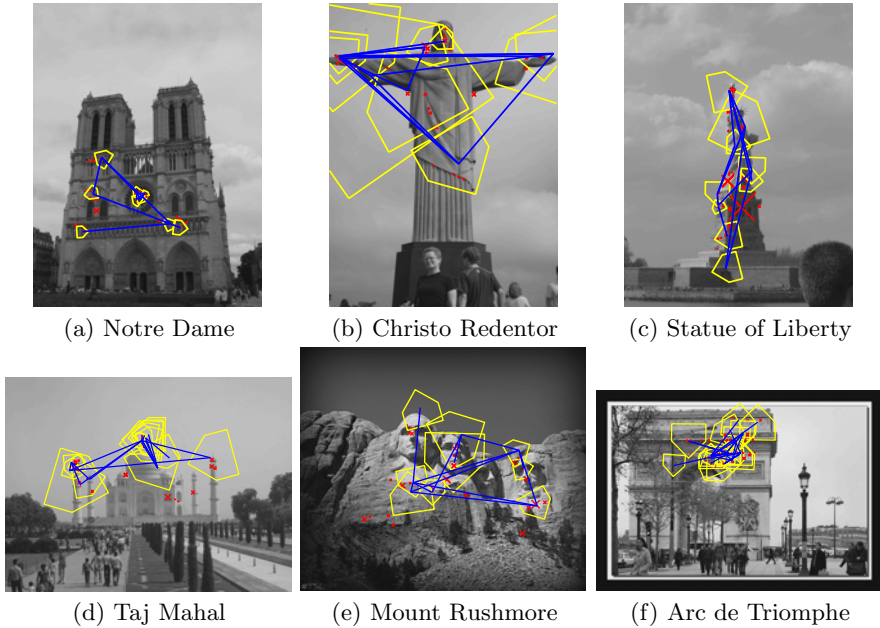
**Fig. 6.** A comparison of precision–recall curves over the HOCKEY (a) and LANDMARK test sets, for three systems: MPMs with our new initialization, our new initialization alone and the system described in [1]. Our initialization system substantially improves overall recall in both image sets. MPMs have little effect in the HOCKEY set, where the distinctive portions of a player’s appearance are of limited size and do not tend to co-occur in repeating patterns. In contrast, MPMs significantly improve precision for the LANDMARK set, perhaps because distinctive portions of landmarks more often co-occur with stable spatial relationships.

## 6.2 Experiments on the LANDMARK Data Set

The LANDMARK data set includes images of 27 famous buildings and locations with some associated tags downloaded from the Flickr website, and randomly divided into 2172 training and 1086 test image–caption pairs. Like the NHL logos, each landmark appears in a variety of perspectives and scales. Compared to the hockey logos, the landmarks usually cover more of the image and have more textured regions in a more stable configuration. On the other hand, the appearance of the landmarks can vary greatly with viewpoint and lighting, and many of the landmarks feature interior as well as exterior views.

Figure 7 provides some sample detections of multipart models in the LANDMARK test set. The MPMs can integrate widely-separated part detections, thereby improving detection confidence and localization. However, many of the models still display a high degree of part overlap, especially on objects such as the Arc de Triomphe with a dense underlying array of distinctive features. In addition, MPM coverage of the object, while better than individual parts, is not as extensive as it could be. For instance, the system detects many more parts on the western face of Notre Dame than are incorporated into the displayed MPM. In the future, we may wish to modify the MPM training routine to explicitly reward spatial coverage improvements. Finally, MPMs often seem to have one or two key parts with a large number of long-range edges. This edge structure may unnecessarily hamper robustness to occlusion.

Regardless of their limitations, Figure 6(b) indicates that MPMs can significantly improve annotation precision. The new initialization system improves overall recall by about 10%, and the addition of MPMs lifts the precision of the curve towards the 100% boundary. The structures on which our system achieved



**Fig. 7.** Sample detections of objects in the LANDMARKS test set

the poorest results were St. Peter’s Basilica, Chichen Itza and the Sydney Opera House. The first two of these suffer from a multiplicity of viewpoints, with training and test sets dominated by a variety of interior viewpoints and zoomed images of different parts of the structure. The Sydney Opera House’s expressionist design has relatively little texture and is therefore harder to recognize using local appearance features.

## 7 Conclusions

Our initialization method and multipart models are designed to work together to improve annotation accuracy and object localization over the approach in [1]. Our initialization mechanism boosts recall and part coverage by detecting potential parts that would have been overlooked by the system in [1], providing for a better distribution of parts over the image set and including more individually ambiguous parts. The MPM layer boosts precision and localization by integrating parts that may be individually ambiguous into models that can cover an entire view of an object.

Together, our new methods significantly improve annotation accuracy over previous results on the experimental data sets, with the amount of improvement strongly dependent on the image set. Our improvements to part initialization and training have significantly increased recall, though sometimes at the expense of precision. For objects with recurring patterns of distinctive parts, the MPM layer



can filter out bad detections, resulting in a substantially improved precision–recall curve.

Our initialization mechanism and the development of multipart models also improves object localization. Parts have less spatial overlap than in [11], they cover portions of the object that are less individually distinctive and they are better-distributed across object views. MPMs tie together recurring patterns of parts, allowing us to distinguish between the presence of multiple parts and multiple objects. Future work could further improve localization by ensuring that MPMs use all available parts to maximize spatial coverage and are themselves well-distributed across object views.

## References

1. Jamieson, M., Fazly, A., Dickinson, S., Stevenson, S., Wachsmuth, S.: Using language to learn structured appearance models for image annotation. *IEEE PAMI* 32, 148–164 (2010)
2. Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D., Jordan, M.: Matching words and pictures. *Journal of Machine Learning Research* 3, 1107–1135 (2003)
3. Carneiro, G., Chan, A., Moreno, P., Vasconcelos, N.: Supervised learning of semantic classes for image annotation and retrieval. *IEEE PAMI* 29, 394–410 (2007)
4. 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)
5. Monay, F., Gatica-Perez, D.: Modeling semantic aspects for cross-media image indexing. *IEEE PAMI* 29, 1802–1817 (2007)
6. Quattoni, A., Collins, M., Darrell, T.: Learning visual representations using images with captions. In: *CVPR* (2007)
7. Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A.: Learning object categories from Google’s image search. In: *CVPR* (2005)
8. Crandall, D.J., Huttenlocher, D.P.: 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)
9. Kokkinos, I., Yuille, A.: HOP: Hierarchical object parsing. In: *CVPR* (2009)
10. Zhu, L., Lin, C., Huang, H., Chen, Y., Yuille, A.: Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part II*. LNCS, vol. 5303, pp. 759–773. Springer, Heidelberg (2008)
11. Bouchard, G., Triggs, B.: Hierarchical part-based visual object categorization. In: *CVPR* (2005)
12. Fidler, S., Boben, M., Leonardis, A.: Similarity-based cross-layered hierarchical representation for object categorization. In: *CVPR* (2008)
13. Epshtein, B., Ullman, S.: Feature hierarchies for object classification. In: *ICCV* (2005)
14. Ommer, B., Buhmann, J.: Learning the compositional nature of visual object categories for recognition. *IEEE PAMI* 32, 501–516 (2010)
15. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *IJCV* 60, 91–110 (2004)
16. Ke, Y., Sukthankar, R.: PCA-SIFT: A more distinctive representation for local image descriptors. In: *CVPR* (2004)

# Voting by Grouping Dependent Parts

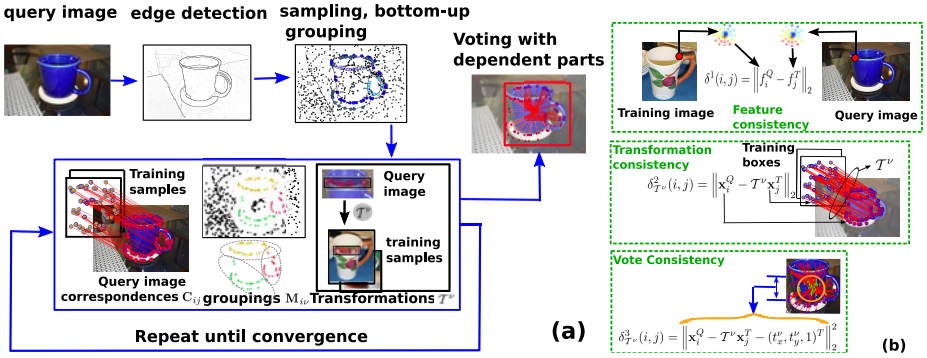
Pradeep Yarlagadda, Antonio Monroy, and Björn Ommer

Interdisciplinary Center for Scientific Computing, University of Heidelberg, Germany  
{pyarlaga, amonroy, bommer}@iwr.uni-heidelberg.de

**Abstract.** Hough voting methods efficiently handle the high complexity of multi-scale, category-level object detection in cluttered scenes. The primary weakness of this approach is however that mutually *dependent local* observations are *independently* voting for intrinsically *global* object properties such as object scale. All the votes are added up to obtain object hypotheses. The assumption is thus that object hypotheses are a sum of independent part votes. Popular representation schemes are, however, based on an overlapping sampling of semi-local image features with large spatial support (e.g. SIFT or geometric blur). Features are thus mutually dependent and we incorporate these dependences into probabilistic Hough voting by presenting an objective function that combines three intimately related problems: i) grouping of mutually dependent parts, ii) solving the correspondence problem conjointly for dependent parts, and iii) finding concerted object hypotheses using extended groups rather than based on local observations alone. Experiments successfully demonstrate that state-of-the-art Hough voting and even sliding windows are significantly improved by utilizing part dependences and jointly optimizing groups, correspondences, and votes.

## 1 Introduction

The two leading methods for detecting objects in cluttered scenes are voting approaches based on the Hough transform [19] and sliding windows (e.g. [33][12]). In the latter case, rectangular sub-regions of a query image are extracted at all locations and scales. A binary classifier is evaluated on each of these windows before applying post-processing such as non-max suppression to detect objects. The computational complexity of this procedure is critical although techniques such as interest point filtering, cascade schemes [33], or branch-and-bound [20] have been presented to address this issue. Rather than using a single, global descriptor for objects, Hough voting avoids the complexity issues by letting local parts vote for parametrized object hypotheses, e.g. object locations and scales. Generalizations of the Hough transform to arbitrary shapes, exemplar recognition [23], and category-level recognition [22][16][29][30][28][25][18] have successfully demonstrated the potential of this approach, and its wide applicability. Despite the current popularity of the method, Hough voting has two significant weaknesses that limit its performance: i) (semi-)local parts are *independently* casting their votes for the object hypothesis and ii) intrinsically global object properties such as object scale [28] have to be estimated locally. Consequently, current voting approaches to object detection, e.g. [22][16][25][18], are adding all local votes in a Hough accumulator and are, thus, assuming that *objects are a sum of their parts*. This assumption is against the fundamental conviction of Gestalt theory that the whole object is more than the sum of its



**Fig. 1.** a) Outline of the processing pipeline. b) The three terms of the cost function  $d_{T^\nu}$  from Eq. (7).

parts. And indeed, popular semi-local feature descriptors such as SIFT [23] or geometric blur [5] have a large spatial support so that different part descriptors in an image are overlapping and thus mutually dependent. To avoid missing critical image details, a recent trend has been to even increase sampling density which entails even more overlap. However, observing the same image region  $N$  times does not provide  $N$  independent estimates of the object hypothesis. Models with richer part dependencies (see section 2) such as constellation models [15] or pictorial structures [14] have been proposed to address these issues, however these methods are limited by their complexity (number of parts and the number of parameters per part). Without grouping, [5] transform a complete query image onto a training image. Therefore, this method is constrained to few distractors (e.g. little background clutter) and the presence of only one object in an image. In [16] Hough voting precedes the complex transformation of the complete object from [5] to limit the hypothesis space and reduce the influence of background clutter. However, the voting is limited by assuming independent part votes.

To establish reliable group votes, we incorporate dependencies between parts into Hough voting [22] by

- *grouping* mutually dependent parts,
- solving the *correspondence problem* (matching parts of the query image to model parts of training images) jointly for all dependent parts, thereby utilizing their information on each other,
- letting groups of dependent parts *vote* for concerted object hypotheses that all constituents of the group agree upon,
- integrating grouping, correspondence, and voting into a single objective function that is *jointly optimized*, since each subtask is depending on the remaining ones.

**Outline of the Approach**

Object detection in a novel image (c.f. Fig. 1) starts by first computing a probabilistic edge map (using [24]). A uniform sampling of edge pixels yields points where local features are extracted on a single scale (we use geometric blur features [5]). Each descriptor is mapped to similar features from training images. In standard Hough voting, all points

are then independently voting for an object hypothesis in scale space, i.e. object location and scale, before adding up all these votes in a Hough accumulator. Consequently, dependencies between points are disregarded and for each point, unreliable local estimates of global object properties such as object scale are required. To correctly model the dependencies between features, we group related points and estimate object hypotheses jointly for whole groups rather than independently for all of their constituents. This results in three intimately related problems: i) Grouping mutually dependent points, ii) letting groups of dependent points vote for a concerted object hypothesis, and iii) finding correspondences for each point in a group to training samples. We jointly find a solution to all of these three subtasks by formulating them in a single cost function and solving it using a single clustering algorithm. That way, all related points influence each others voting and correspondences and their voting influences their grouping, in turn. To obtain an initial grouping, we perform pairwise clustering of edge points. The necessary pairwise affinities are obtained by measuring the cooccurrence of points in different levels of the hierarchical segmentation of the initial probabilistic edge map from [24].

## 2 Voting Methods and Object Detection

Category-level object detection requires models that represent objects based on local measurements in an image. A broad variety of models with widely differing representation complexity have been proposed. These range from bag-of-features approaches [11] and latent topic models without spatial relationships [31] to richer spatial representations such as hierarchical models [7][17][2], k-fans [10], and latent scene models [32]. Complex spatial representations have been described by a joint model of all local parts (constellation model) [15], shape matching [5], pictorial structures [14], and by rigid template-like models [12][21]. The compositional nature of our visual world has been utilized by [27] to build hierarchical object representations. [26] describes a Tensor voting approach to form perceptually meaningful groups which can then be used for object recognition. The voting paradigm [22][16][28][25][18], which is central to this paper, effectively handles the complexity of large-scale part-based models.

### 2.1 Hough Voting with Independent Parts

Hough voting makes part-based object models with large numbers of parts feasible by letting all parts independently cast their votes for object hypotheses [22]. All these locally estimated object hypotheses are summed up in a Hough accumulator  $\mathcal{H}^{\text{pnt}}(c, \mathbf{x}, \sigma)$  over scale space. Here,  $\mathbf{x}$  and  $\sigma$  are the location and scale of an object hypothesis and  $c$  denotes its category. Moreover, a local part detected at location  $\mathbf{x}_i^Q \in \mathbb{R}^2$  in a query image incorporates a feature vector  $f_i^Q \in \mathbb{R}^N$  and a local estimate  $\sigma_i^Q \in \mathbb{R}$  of object scale. The key assumption of Hough voting is that all parts are *independently* casting their votes for the object hypothesis so that the overall *object hypothesis is independently obtained from dependent parts*,

$$\mathcal{H}^{\text{pnt}}(c, \mathbf{x}, \sigma) \propto \sum_i p(\mathbf{x}, \sigma | c, f_i^Q, \mathbf{x}_i^Q, \sigma_i^Q) p(c | f_i^Q, \mathbf{x}_i^Q, \sigma_i^Q) \quad (1)$$

Let  $f_j^T$  denote the  $j$ -th codebook vector or the  $j$ -th training sample, depending on whether vector quantization or a nearest neighbor approach is used. Without loss of generality we can assume that the training object is centered at the origin so that the location  $\mathbf{x}_j^T \in \mathbb{R}^2$  of  $f_j^T$  is the shift of the feature from the object center. Moreover, all training images are assumed to be scale normalized, i.e. they are rescaled so that objects are the same size. Summation over  $f_j^T$  and  $\mathbf{x}_j^T$  then yields

$$\mathcal{H}^{\text{pnt}}(c, \mathbf{x}, \sigma) \propto \sum_{i,j} p(\mathbf{x} - [\mathbf{x}_i^Q - \sigma_i^Q \mathbf{x}_j^T], \sigma - \sigma_i^Q) \times p(c|f_j^T) p(f_j^T|f_i^Q) \quad (2)$$

Details of this derivation can be found in [22][28].

## 2.2 Key Points of Our Method

Hough voting methods (e.g. [22][16][28][25][18]) let all parts independently cast their votes for the object hypothesis, thereby neglecting part dependence. In contrast to this, our approach models the dependencies between parts by establishing groups and letting all parts in a group jointly find a concerted object hypothesis. In detail, we are differing from voting methods to detection in the following ways:

**Grouping of Dependent Parts:** Rather than considering all parts to provide independent votes (e.g. [22][16][28][25][18]), we segment a scene into groups of mutually dependent parts. Thus multiple strongly related features (e.g. due to overlapping descriptors) are not considered as providing independent information.

**Joint Voting of Groups of Dependent Parts:** Mutually dependent parts in a group assist each other in finding compatible correspondences and votes, rather than estimating these independently as in standard Hough voting. Thus groups yield votes with significantly less uncertainty than the individual part votes (c.f. Fig. 5). Intrinsically global parameters such as object scale are then obtained by global optimization rather than by local estimates (such as local scale estimation in [22][8]). [28] could only model the uncertainty of each local part. Based on a grouping of parts, we can however obtain reliable estimates.

**Joint Optimization of Grouping, Voting, and Correspondences:** Identifying and grouping dependent parts, computing joint votes for complete groups, and solving the part correspondence problem are mutually dependent problems of object detection. We tackle them jointly by iteratively optimizing a single objective function. Rather than letting each of these factors influence the others, [8] finds groups before using them to optimize correspondences in a model where parts are grouped with their  $k$  nearest neighbors. Estrada et al. [13] pursue the simpler problem of exemplar matching by only dealing with grouping and matching consecutively. Several extensions have been proposed to the standard Hough voting scheme, but the critical grouping of dependent parts has not been integrated into voting in any of those approaches. [29] extend the Implicit Shape Model by using curve fragments as parts that cast votes. Without incorporating a grouping stage into their voting, parts are still independently casting their votes. Amit et al. [3] propose a system limited to triplet groupings. In contrast to such rigid

groupings, our approach combines flexible numbers of parts based on their vote consistency and geometrical distortion. In contrast to hierarchical grouping approaches, where later groupings build on earlier ones, our method does not require any greedy decisions that would prematurely commit to groupings in earlier stages but rather optimizes all groupings at the same time.

**Linear Number of Consistency Constraints:** In contrast to Berg et al. [5] who need a quadratic number of consistency constraints between all pairs of parts, grouping reduces this to a linear number of constraints between parts and the group they belong to, see section 3.

**Flexible Model vs. Rigid Template:** Template-like descriptors such as HoG [12] or [21] have a rigid spatial layout that assumes objects to be box-shaped and non-articulated. Moreover, they require a computationally daunting search through hypothesis space although approximations such as branch-and-bound [20] have been proposed to deal with this issue. On the other end of the modeling spectrum are flexible parts-and-structure models [15][14]. However, the modeling of part dependencies in [15] becomes prohibitive for anything but very small number of points and [14] restrict the dependencies to a single, manually selected reference part. In contrast to this, we incorporate dependencies in the powerful yet very efficient Hough voting framework. Moreover, we do not rely on pixel accurate labeling of foreground regions as in [22] but only utilize bounding box annotations. In contrast to [16][5] who transform a query image onto training images using a complex, nonlinear transformation we decompose the object and the background into groups and transform these onto the training samples using individual, linear transformations. That way, unrelated regions do not interfere in a single, complex transformation and regions of related parts can be described by simpler and thus more robust, linear models.

### 3 Grouping, Voting, and Correspondences

Hough voting approaches to object detection let all local parts independently vote for a conjoint object hypothesis. However, there are direct mutual dependencies between features, e.g. due to their large spatial support and since interest point detection has a bias towards related regions in background clutter [6]. Thus, multiple related features yield dependent votes rather than independent evidence on the object. Rather than adding up all those duplicates as is common practice in Hough voting approaches (eg. [22][16][25][28]), a group of mutually dependent parts should actually jointly vote for a concerted object hypothesis. That way, the correspondence problem of matching features in a novel query image to features in training samples is jointly solved for a group of dependent parts.

#### 3.1 Joint Objective Function for Grouping, Voting, and Correspondences

To solve the grouping, voting, and correspondence problem jointly, we have to i) match query features onto related training features, ii) find correspondences with low geometrical distortion, and iii) minimize the overall scatter of all votes within a group. Let us

now investigate each of these aspects in detail. Hough voting solves the correspondence problem by matching the  $i$ -th part of a query image,  $f_i^Q$ , to the training part or training codebook vector  $f_j^T$  that is most similar, i.e. for which

$$\delta^1(i, j) = \left\| f_i^Q - f_j^T \right\|_2 \tag{3}$$

is minimal. Boiman et al. [6] have demonstrated the deficiencies of quantization and codebook based representations. Therefore, we adopt a nearest neighbor approach, where query features are mapped onto training features rather than mapping them onto a quantized codebook. Let  $C_{ij} \in \{0, 1\}$  denote a matching of the  $i$ -th query part to the  $j$ -th training part, where  $C_{ij}$  captures many-to-one-matchings,  $\sum_j C_{ij} = 1$ . As discussed above, the correspondence problem has to be solved jointly for all mutually dependent parts, i.e. all related parts should undergo the same transformation  $T^\nu$  when being matched to the training samples,  $\mathbf{x}_i^Q \stackrel{!}{=} T^\nu \mathbf{x}_j^T$ . This implies that related parts  $i$  and  $i'$  are clustered into the same group  $\nu$  by computing assignments  $M_{i\nu}$  of parts to groups,  $M_{i\nu} \in \{0, 1\}$ ,  $\sum_\nu M_{i\nu} = 1$ .

Due to the relatedness of points in a group, transformations should be forced to be simple, eg. similarity transformations

$$T^\nu = \begin{pmatrix} \sigma_x^\nu \cos(\theta) & -\sigma_y^\nu \sin(\theta) & t_x^\nu \\ \sigma_x^\nu \sin(\theta) & \sigma_y^\nu \cos(\theta) & t_y^\nu \\ 0 & 0 & 1 \end{pmatrix} \tag{4}$$

In effect, we are decomposing heterogeneous objects into groups of dependent parts so that piecewise linear transformations (one for each group) are sufficient rather than using a complex nonlinear transformation for the whole scene as in [5][16]. Let  $G^\nu := \{i : M_{i\nu} = 1\}$  denote all parts in a group  $\nu$  and  $|G^\nu| = \sum_i M_{i\nu}$  denote the number of parts in the group. Then we have to find a transformation  $T^\nu$  that minimizes the distortion

$$\delta_{T^\nu}^2(i, j) = \left\| \mathbf{x}_i^Q - T^\nu \mathbf{x}_j^T \right\|_2 \tag{5}$$

for each part in the group.

[5] is penalizing the distortions of correspondences to yield minimal group distortion. The consistency of group votes is obtained by measuring the deviation of individual votes from the average vote of the group. Minimal group distortion does not necessarily guarantee consistent group votes. Hence we introduce a term that penalizes the scatter of the group vote.

$$\delta_{T^\nu}^3(i, j) = \left\| \mathbf{x}_i^Q - T^\nu \mathbf{x}_j^T - (t_x^\nu, t_y^\nu, 1)^T \right\|_2^2 \tag{6}$$

[6] is measuring the agreement of all parts in the group with respect to their object center estimate (summing over all parts  $i$  in a group yields the variance of the group vote).

This consistency constraint has a linear complexity in the number of image features in contrast to Berg et al. [5] who proposed pairwise consistency constraints with a quadratic complexity. This reduction in complexity is possible since dependent parts are combined in groups, so we can penalize the scatter of the entire group. Without the grouping, Berg et al. have to penalize the distortions of all pairs of parts under the transformation.

### Joint Cost Function

Groupings  $\mathbf{M}_{i\nu}$  of query parts, correspondences  $\mathbf{C}_{ij}$  between query parts and training parts, and group transformations  $T^\nu$  are mutually dependent. Thus we have to combine them in a single cost function

$$d_{T^\nu}(i, j) = \lambda_1 \delta^1(i, j) + \lambda_2 \delta_{T^\nu}^2(i, j) + \lambda_3 \delta_{T^\nu}^3(i, j) \quad (7)$$

that is jointly optimized for each of these unknowns. The weights  $\lambda_1, \lambda_2, \lambda_3$  are adjusted by measuring the distribution of each distance term  $\delta(\cdot)$  in the training data. The weights are then set to standardize the dynamic range of each term to the same range. The cost for matching all the query parts  $i$  which belong to group  $\nu$  to the corresponding training parts  $j = \mathbf{C}(i)$  is given by

$$\mathcal{R}(G^\nu) = \frac{1}{|G^\nu|} \sum_i \mathbf{M}_{i\nu} \sum_j \mathbf{C}_{ij} d_{T^\nu}(i, j) \quad (8)$$

### 3.2 Joint Optimization of Groups, Votes, and Correspondences

To find optimal groups, object votes, and correspondences, we need to minimize the overall cost of all groups  $\sum_\nu \mathcal{R}(G^\nu)$ . We seek optimal group assignments  $\mathbf{M}^*$ , correspondences  $\mathbf{C}^*$ , and transformations  $T^*$  that minimize the summation of costs over all the groups,

$$(\mathbf{M}^*, \mathbf{C}^*, T^*) = \operatorname{argmin}_{\mathbf{M}, \mathbf{C}, T} \sum_\nu \mathcal{R}(G^\nu). \quad (9)$$

Since parts in a group are mutually dependent, each of these parameters depends on the other two. Therefore we incorporate an alternating optimization scheme. To find the optimal corresponding training part  $j = \mathbf{C}(i)$  for query part  $i$  we have to minimize

$$\mathbf{C}(i) = \operatorname{argmin}_j d_{T^\nu}(i, j). \quad (10)$$

So for each  $i$ , we select the training part  $j$  with minimal cost. Optimal groupings are obtained by finding assignments  $\nu = \mathbf{M}_{i\nu}(i)$  for each part  $i$ ,

$$\mathbf{M}_{i\nu}(i) = \operatorname{argmin}_\nu d_{T^\nu}(i, \mathbf{C}(i)) = \operatorname{argmin}_\nu [\lambda_2 \delta_{T^\nu}^2(i, \mathbf{C}(i)) + \lambda_3 \delta_{T^\nu}^3(i, \mathbf{C}(i))]. \quad (11)$$



Thus for each  $i$ , the group  $\nu$  with minimal distortion is chosen. Finally, the transformation of each group from the query image onto the training images has to be estimated

$$\mathcal{T}^\nu = \underset{\mathcal{T}}{\operatorname{argmin}} \sum_i \mathbf{M}_{i\nu} \sum_j \mathbf{C}_{ij} \cdot [\lambda_2 \delta_{\mathcal{T}^\nu}^2(i, \mathbf{C}(i)) + \lambda_3 \delta_{\mathcal{T}^\nu}^3(i, \mathbf{C}(i))] . \quad (12)$$

Optimal  $\mathcal{T}^\nu$  in (12) is obtained by Levenberg-Marquardt minimization. These three optimization steps are alternated until convergence. In our experiments, the optimization in Alg. 1 has usually converged after two or three iterations. We initialize  $\nu$  by the output of a bottom-up grouping that is outlined in section 3.4. Initialization of  $\mathbf{C}_{ij}$  for each query part  $i$  is obtained by a nearest neighbour search for  $j$  using the distance function  $\delta^1(i, j)$ .  $\mathcal{T}^\nu$  is initialized with the transformation that aligns the centroid of group  $\nu$  onto the centroid of the corresponding training parts.

### 3.3 Hough Voting with Groups

After finding optimal groupings, group transformations, and correspondences, the votes from all groups have to be combined. In standard Hough voting, the votes of all parts are summed up, thus treating them as being independent, c.f. the discussions in [34, 1]. In our setting, all mutually dependent parts are combined in the same group. The joint optimization of correspondences and transformations forces these dependent parts to agree upon a joint overall vote.

$$(\mathbf{x}, \sigma)^\top = (\mathbf{x}_i^Q - \mathcal{T}^\nu \mathbf{x}_j^T \mathbf{C}(i) + t^\nu, \sigma^\nu)^\top \quad (13)$$

where  $t^\nu$  and  $\sigma^\nu$  are the translation and scaling component of  $\mathcal{T}^\nu$ . Evidently, all parts in a group are coupled by using the same transformation matrix  $\mathcal{T}^\nu$  and the jointly optimized correspondences  $\mathbf{C}_{ij}$ . After jointly optimizing the votes of all dependent parts, the group vote can be obtained by averaging over the part votes. The Hough accumulator for the voting of groups is obtained by summing over independent groups rather than over dependent parts as in standard Hough voting. Since groups are mutually independent, their summation is justified. Analogous to (2) we obtain

$$\mathcal{H}^{\text{grp}}(c, \mathbf{x}, \sigma) \propto \sum_\nu \frac{1}{G^\nu \mathcal{R}(G^\nu)} \times \sum_{i \in G^\nu} \sum_j \mathbf{C}_{ij} \cdot P(\mathbf{x} - [\mathbf{x}_i^Q - \mathcal{T}^\nu \mathbf{x}_j^T + t^\nu], \sigma - \sigma^\nu) \quad (14)$$

where  $P(\bullet)$  is obtained using the balloon density estimator [9] with Gaussian Kernel  $K$ , Kernel bandwidth  $b$ , and distance function in scale space  $d : \mathbb{R}^3 \times \mathbb{R}^3 \mapsto \mathbb{R}$ ,

$$P(\mathbf{x} - [\mathbf{x}_i^Q - \mathcal{T}^\nu \mathbf{x}_j^T + t^\nu], \sigma - \sigma^\nu) = K \left( \frac{d \left[ (\mathbf{x}, \sigma)^\top; (\mathbf{x}_i^Q - \mathcal{T}^\nu \mathbf{x}_j^T + t^\nu, \sigma^\nu)^\top \right]}{b(\sigma)} \right) \quad (15)$$

---

**Algorithm 1.** Voting with groups of dependent parts: Joint optimization of groupings, correspondences, and transformations.

---

Input: • parts from query image:  $f_i^Q, \mathbf{x}_i^Q$ ,  
 • UCM-connectivity [4]  $\bar{\mathbf{A}}_{ii'}$   
 • parts from all training images:  $f_j^T, \mathbf{x}_j^T$   
 Init: • pairwise clustering on  $\bar{\mathbf{A}}_{ii'} \rightarrow \mathbf{M}_{iiv}()$

- 1 **do**
- 2    $\mathbf{C}(i) \leftarrow \operatorname{argmin}_j d_{T^\nu}(i, j)$
- 3    $\mathbf{M}_{iiv}(i) \leftarrow \operatorname{argmin}_v d_{T^\nu}(i, \mathbf{C}(i))$
- 4    $T^\nu \leftarrow \operatorname{argmin}_T \sum_i \mathbf{M}_{iiv} \sum_j \mathbf{C}_{ij} (\lambda_2 \delta_{T^\nu}^2(i, \mathbf{C}(i)) + \lambda_3 \delta_{T^\nu}^3(i, \mathbf{C}(i)))$
- 5 **until** convergence
- 6  $\mathcal{H}^{\text{SP}}(c, \mathbf{x}, \sigma) \leftarrow \text{Eq. (14)}$
- 7  $\{(\mathbf{x}^h, \sigma^h)^\top\}_h \leftarrow \text{Local minima of } \mathcal{H}^{\text{SP}}$

---

### 3.4 Bottom-Up Grouping

Object detection in a query image starts by computing a probabilistic edge map [4] and uniformly sampling edge points. Next, we perform a bottom-up grouping on the probabilistic edges which serves as an initialization for  $\nu$  in section 3.1. Two edge points  $i, i'$  are considered to be connected on level  $s$  of the hierarchical ultrametric contour map of [4], if they are on the boundary of the same region on this level. Let  $1 = \mathbf{A}_{ii'}^s \in \{0, 1\}$  denote this case. Averaging over all levels,  $\bar{\mathbf{A}}_{ii'} \propto \sum_s \mathbf{A}_{ii'}^s$ , yields a similarity measure between points and pairwise clustering (using Ward’s method) on this similarity matrix produces a grouping  $\mathbf{M}_{iiv}$  which we use to initialize the optimization of (9).

### 3.5 Hypothesis Verification

Due to intra-class variations and noise, the votes of all parts in a group cannot be brought into perfect agreement. As is common practice in voting approaches, we employ a verification stage, where a SVM classifies histograms of oriented gradients (extracted on regular grids on 4 different resolutions and 9 orientations) using pyramid match kernels (PMK). To train the SVM, positive examples for a category are the groundtruth bounding boxes, rescaled to the average bounding box diagonal length of the class. Negative samples are obtained by running our group voting on the positive training samples and selecting false positive hypotheses, i.e. the most confused negative samples. In the verification stage, the SVM classifier is evaluated in a local  $3 \times 3$  neighbourhood around each voting hypothesis. This local search refines the voting hypotheses from the groups.

## 4 Experiments

We evaluate our approach on ETHZ Shape and INRIA Horses Datasets. These two datasets feature significant scale changes, intra-class variation, multiple-objects per image, and intense background clutter. We use the latest experimental protocol of Ferrari et al. [16]: For ETHZ shape dataset, detectors are trained on half the positive samples of

a category. No negative training images are used and all remaining images are used for testing. For INRIA shape dataset, 50 horse images are used for training and the remaining 120 horse images plus 170 negative images are used for testing. In all experiments, the detection performance is measured using the PASCAL VOC criterion [16] (requiring the ratio of intersection and union of predicted and groundtruth bounding box to be greater than .5).

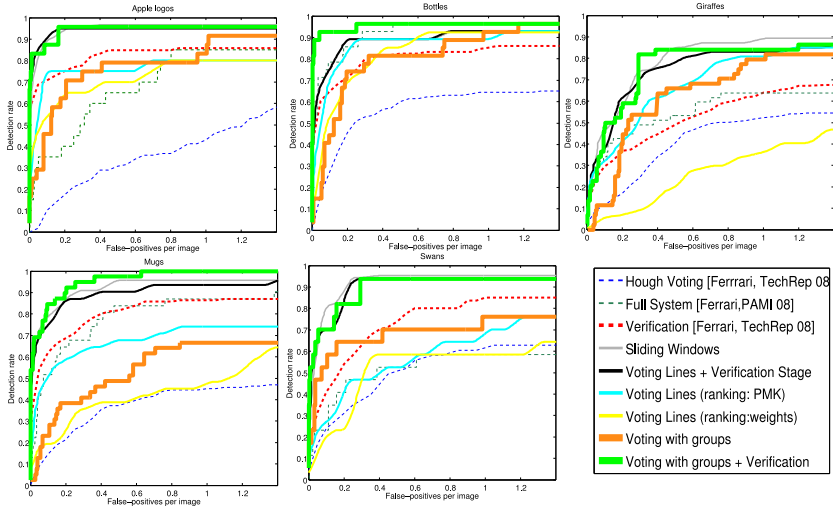
#### 4.1 ETHZ Shape Dataset – Performance Analysis

Fig. 2 compares our approach with state-of-the-art voting methods on ETHZ. Voting with our groups of dependent parts outperforms all current voting based approaches. We achieve a gain of 27% over the Hough voting in [16], an improvement of 19% over [25], and 17% higher performance than [28], see Tab. 1. Even compared with the local sliding window classification in [28] (PMK re-ranking) we obtain a slightly higher performance (1.4%). The PMK re-ranking is a separate classifier that performs verification of votes. Thus our voting method alone not only improves current Hough voting approaches, but also produces results beyond those of the verification stage of some of the methods.

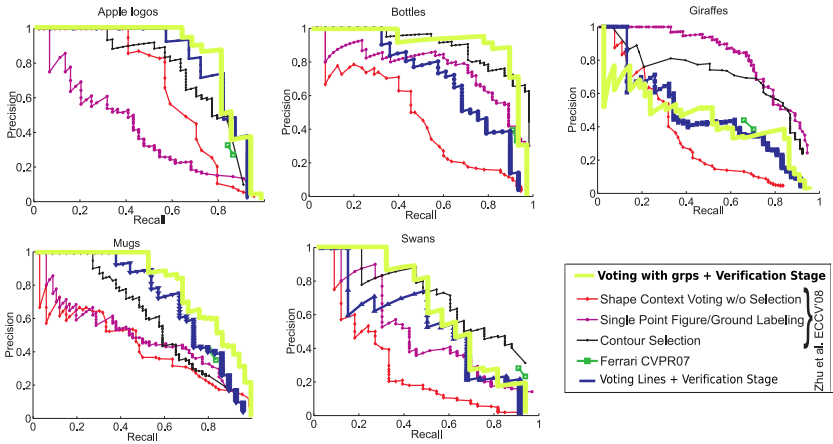
The primary focus of this paper is to improve Hough voting by modeling part dependence. Nevertheless, we also investigate the combined detector consisting of voting and a verification stage. The results are shown in Fig. 2. Our results compare favourably with sliding window classification in [28]. This approach has to search over  $10^4$  hypotheses whereas our approach produces on the order of 10 candidate hypotheses. Consequently, the gain in computational performance of our approach is between two and three orders of magnitude. Compared to preprocessing steps such as extraction of probabilistic edge maps and computation of geometric blur, our grouping, voting and correspondence optimization has insignificant running time. Nevertheless, we obtain a gain of 3.68% over sliding windows at 0.3 fppi. Compared to the best verification systems [25], we obtain a gain of 0.68% at 0.3 fppi.

Fig. 3 compares the supervised methods of [35] against our detector (which only needs training images with bounding boxes). Without requiring the supervision information of [35], we are dealing with a significantly harder task. [16] showed a performance loss of 15% at 0.4 fppi. Nevertheless, we perform better on 3 out of 5 categories. (actual values of [35] are unavailable).

Let us now compare the reliability of votes from individual parts with the reliability of object hypotheses produced by our groupings. Therefore, we map object query features (features from within the groundtruth bounding box) onto the positive training samples and we do the same for background query features. By comparing the matching costs we see how likely positive query features are mistaken to be background and vice versa. Then we are doing the same for groups, i.e. groupings (11) from the object and from the background are mapped onto positive training samples. Fig. 5 shows that groups have a significantly lower error rate  $\mathcal{R}$  (30% vs. 77%) to be mapped onto wrong training samples. Thus group votes are significantly more reliable. Fig. 4 shows the voting of parts before and after optimization. Voting with groups produces concerted votes whereas independent parts (singleton groups) produce votes with significant clutter.



**Fig. 2.** Detection performance. On average our voting approach yields a 27% higher performance than standard Hough voting and improves line voting [28] by 17%.



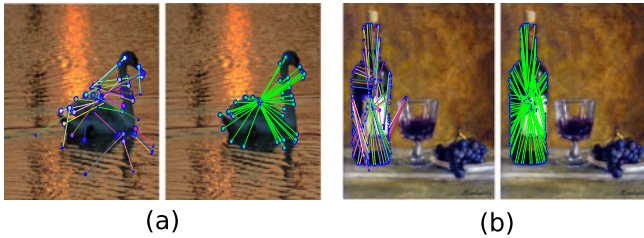
**Fig. 3.** Comparing our voting+verification with the supervised approach [35]. [16] has shown that our training scenario is significantly harder and yields 13% lower recall at .4 FPPI

### 4.2 INRIA Horse Dataset – Performance Analysis

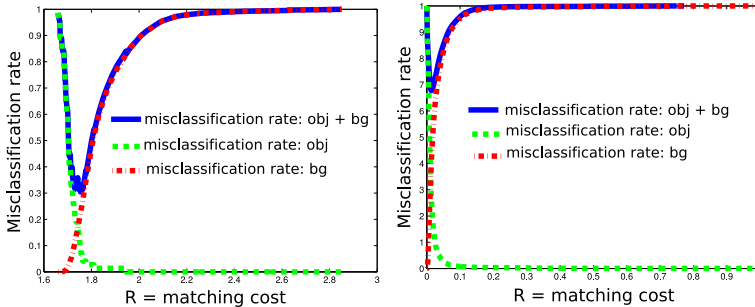
Figure Fig. 6 shows the performance of voting with groups and the overall detector (voting + verification). Voting with groups significantly outperforms the best voting methods so far (M<sup>2</sup>HT detector), e.g., roughly 12% gain at 3 fppi. In terms of overall performance, we have a detection rate of 87.3% at 1 fppi compared to the state of the art results of 85.27% for M<sup>2</sup>HT + IKSVM and 86% for sliding windows (IKSVM).

**Table 1.** Comparing the performance of various methods. Detection rates (in [%]), PASCAL criterion .5 overlap. The approach of [25] use positive as well as negative samples for training whereas we use only positive samples for training. Our voting yields a 27% higher performance than the Hough voting in [16], 19% gain over max-margin Hough voting [25], and 17% gain over line voting [28], thus significantly improving the state-of-the-art in voting.

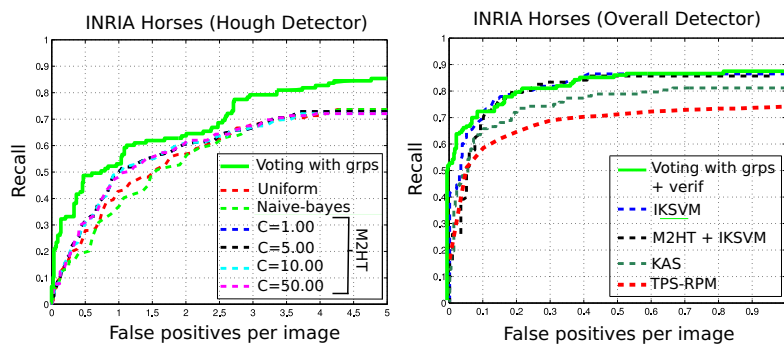
Cat	Voting Stage (FPPI = 1.0)				Verification Stage (FPPI = 0.3 / 0.4)				
	$\mathcal{H}^{gfp}$	Hough [16]	M <sup>2</sup> [25]	HT voting [28]	$\mathcal{H}^{gfp}$ vo- ting+verif	Full system [28]	Sliding Windows	Full syst [16]	M <sup>2</sup> HT+ IKSVM [25]
Apples	84.0	43.0	85.0	80.0	95.83 / 95.83	95.0 / 95.0	95.8 / 96.6	77.7 / 83.2	95.0 / 95.0
Bottles	93.1	64.4	67.0	92.4	96.3 / 96.3	89.3 / 89.3	89.3 / 89.3	79.8 / 81.6	92.9 / 96.4
Giraffes	79.5	52.2	55.0	36.2	81.82 / 84.09	70.5 / 75.4	73.9 / 77.3	39.9 / 44.5	89.6 / 89.6
Mugs	67.0	45.1	55.0	47.5	94.87 / 96.44	87.3 / 90.3	91.0 / 91.8	75.1 / 80.0	93.6 / 96.7
Swans	76.6	62.0	42.5	58.8	94.12 / 94.12	94.1 / 94.1	94.8 / 95.7	63.2 / 70.5	88.2 / 88.2
Avg	80.0	53.3	60.9	63.0	92.58 / 93.35	87.2 / 88.8	88.9 / 90.1	67.2 / 72.0	91.9 / 93.2



**Fig. 4.** Left plot in panels (a) and (b) shows standard Hough voting which assumes mutual independence between features. Right plot in panels (a) and (b) shows the voting after joint optimization of correspondences, groups, and votes.



**Fig. 5.** Reliability of parts (singleton groups), left plot vs. groups, right plot. The plots show the misclassification rate of groups and parts for different matching cost  $\mathcal{R}$ . The optimal error rate for parts is 77%, for groups 30% thereby underlining the increased reliability of groups.



**Fig. 6.** Detection plots on INRIA Horses dataset. Left plot compares the  $M^2HT$  detector for different parameters with our group voting. Voting with groups is superior to all. Right plot compares the overall detection results obtained from voting with groups plus verification with sliding windows (IKSVM) and state-of-the-art methods. At 1.0 FPPI we achieve a detection rate of 87.3% compared to the state of the art result of 86% (IKSVM) [25]

## 5 Discussion

We have tackled the primary weakness of Hough voting methods, the assumption of part independence, by introducing the grouping of mutually dependent parts into the voting procedure. Therefore, we have formulated voting-based object detection as an optimization problem that jointly optimizes groupings of dependent parts, correspondences between parts and object models, and votes from groups to object hypotheses. Rather than using uncertain local votes from unreliable local parts we utilize their dependences to establish extended groups that reliably predict global object properties and are thus producing reliable object hypotheses. Compared to the sliding window paradigm, our voting approach reduces the number of candidate hypotheses by three orders of magnitude and improves its recall. Our model of part dependence in voting has demonstrated that it significantly improves the performance of probabilistic Hough voting in object detection.

**Acknowledgements.** This work was supported by the Excellence Initiative of the German Federal Government, DFG project number ZUK 49/1.

## References

1. Lehmann, B.L.A., van Gool, L.: Prism principled implicit shape model. In: BMVC (2008)
2. Ahuja, N., Todorovic, S.: Connected segmentation tree: A joint representation of region layout and hierarchy. In: CVPR (2008)
3. Amit, Y., Geman, D.: A computational model for visual selection. *Neural Computation* (1999)
4. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: From contours to regions: An empirical evaluation. In: CVPR (2009)
5. Berg, A.C., Berg, T.L., Malik, J.: Shape matching and object recognition using low distortion correspondence. In: CVPR, pp. 26–33 (2005)
6. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: CVPR (2008)
7. Bouchard, G., Triggs, B.: Hierarchical part-based visual object categorization. In: CVPR, pp. 710–715 (2005)

8. Carneiro, G., Lowe, D.: Sparse flexible models of local features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 29–43. Springer, Heidelberg (2006)
9. Comaniciu, D., Ramesh, V., Meer, P.: The variable bandwidth mean shift and data-driven scale selection. In: ICCV, pp. 438–445 (2001)
10. Crandall, D.J., Felzenszwalb, P.F., Huttenlocher, D.P.: Spatial priors for part-based recognition using statistical models. In: CVPR, pp. 10–17 (2005)
11. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: ECCV, Workshop Stat. Learn. in Comp. Vis. (2004)
12. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)
13. Estrada, F.J., Fua, P., Lepetit, V., Susstrunk, S.: Appearance-based keypoint clustering. In: CVPR (2009)
14. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. IJCV 61(1) (2005)
15. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR, pp. 264–271 (2003)
16. Ferrari, V., Jurie, F., Schmid, C.: From images to shape models for object detection. IJCV (2009)
17. Fidler, S., Boben, M., Leonardis, A.: Similarity-based cross-layered hierarchical representation for object categorization. In: CVPR (2008)
18. Gall, J., Lempitsky, V.: Class-specific hough forests for object detection. In: CVPR (2009)
19. Hough, P.: Method and means for recognizing complex patterns. U.S. Patent 3069654 (1962)
20. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Beyond sliding windows: Object localization by efficient subwindow search. In: CVPR (2008)
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. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. IJCV 77(1-3), 259–289 (2008)
23. Lowe, D.: Object recognition from local scale-invariant features. In: ICCV (1999)
24. Maire, M., Arbelaez, P., Fowlkes, C., Malik, J.: Using contours to detect and localize junctions in natural images. In: CVPR (2008)
25. Maji, S., Malik, J.: Object detection using a max-margin hough transform. In: CVPR (2009)
26. Medioni, G., Tang, C., Lee, M.: Tensor voting: Theory and applications. In: RFIA (2000)
27. Ommer, B., Buhmann, J.: Learning the compositional nature of visual object categories for recognition. PAMI 32(3), 501–516 (2010)
28. Ommer, B., Malik, J.: Multi-scale object detection by clustering lines. In: ICCV (2009)
29. Opelt, A., Pinz, A., Zisserman, A.: Incremental learning of object detectors using a visual shape alphabet. In: CVPR, pp. 3–10 (2006)
30. Shotton, J., Blake, A., Cipolla, R.: Contour-based learning for object detection. In: ICCV (2005)
31. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering objects and their localization in images. In: ICCV, pp. 370–377 (2005)
32. Sudderth, E.B., Torralba, A.B., Freeman, W.T., Willsky, A.S.: Learning hierarchical models of scenes, objects, and parts. In: ICCV, pp. 1331–1338 (2005)
33. Viola, P.A., Jones, M.J.: Robust real-time face detection. IJCV 57(2), 137–154 (2004)
34. Williams, C., Allan, M.: On a connection between object localization with a generative template of features and pose-space prediction methods. Technical report, University of Edinburgh, Edinburgh (2006)
35. Zhu, Q.H., Wang, L.M., Wu, Y., Shi, J.B.: Contour context selection for object detection: A set-to-set contour matching approach. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 774–787. Springer, Heidelberg (2008)

# Superpixels and Supervoxels in an Energy Optimization Framework

Olga Veksler, Yuri Boykov, and Paria Mehrani

Computer Science Department, University of Western Ontario  
London, Canada

{olga,yuri,pmehrani}@uwo.ca

<http://www.csd.uwo.ca/faculty/olga/>

**Abstract.** Many methods for object recognition, segmentation, etc., rely on a tessellation of an image into “superpixels”. A superpixel is an image patch which is better aligned with intensity edges than a rectangular patch. Superpixels can be extracted with any segmentation algorithm, however, most of them produce highly irregular superpixels, with widely varying sizes and shapes. A more regular space tessellation may be desired. We formulate the superpixel partitioning problem in an energy minimization framework, and optimize with graph cuts. Our energy function explicitly encourages regular superpixels. We explore variations of the basic energy, which allow a trade-off between a less regular tessellation but more accurate boundaries or better efficiency. Our advantage over previous work is computational efficiency, principled optimization, and applicability to 3D “supervoxel” segmentation. We achieve high boundary recall on images and spatial coherence on video. We also show that compact superpixels improve accuracy on a simple application of salient object segmentation.

**Keywords:** Superpixels, supervoxels, graph cuts.

## 1 Introduction

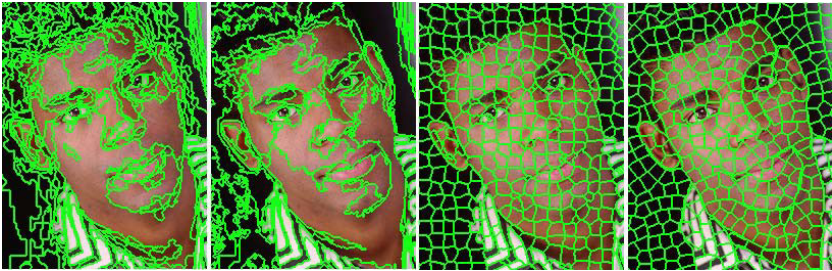
Many vision applications benefit from representing an image as a collection of *superpixels*, for example [1,2,3,4,5,6,7,8], to cite just a few. While the exact definition of a superpixel is not feasible, it is regarded as a perceptually meaningful atomic region. A superpixel should contain pixels that are similar in color, texture, etc., and therefore are likely to belong to the same physical world object. The atomic region notion is old, but a popular term *superpixel* has been coined recently [1].

The assumption that all pixels in a superpixel belong to the same object leads to the advantage of superpixel primitives over pixel primitives. The first advantage is computational efficiency. If one needs to compute a property that stays approximately constant for an object, then superpixel representation is more efficient since the total number of primitives is greatly reduced [9]. Computational efficiency also comes from a reduction in the number of hypothesis. Instead of



exhaustive examining of all rectangular patches [10], an alternative is to examine only superpixels [12,45,6,7]. In addition to efficiency, superpixels are used for computing features that need spatial support [3].

To obtain superpixels, one often uses image segmentation algorithms such as meanshift [11], graph based [12], normalized cuts [13]. To increase the chance that superpixels do not cross object boundaries, a segmentation algorithm is run in an oversegmentation mode. However, most segmentation algorithms produce regions of highly irregular shape and size, for example the meanshift [11] and the graph-based [12], see Fig. 1, first two images. The boundaries are also highly irregular, since there is no explicit constraints on length. A large superpixel with a highly irregular shape is likely to straddle more than one object.



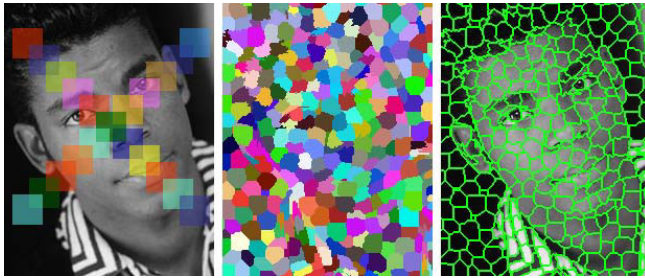
**Fig. 1.** From left to right: meanshift [11], graph based [12], turbopixels [14], NC superpixels [1]. Implementation was obtained from the authors’ web sites.

There are advantages to superpixels with regular shapes and sizes, such as those in Fig. 1, right. A regular shape is less likely to cross object boundaries, since objects rarely have wiggly shapes. If a superpixel does cover more than one object, if its size is not too large, the error rate is likely to be controlled.

The normalized cuts algorithm [13] can be adapted to compute superpixels that are regular in size and shape [1], see Fig. 1. Many methods that need regular superpixels use normalized cuts [9,11,2,4,5]. However, NC superpixels [1] are very expensive, and have the following unappealing property, noticed by [14]. The smaller is the size of target superpixels, the longer the computation takes.

Our work was inspired by the turbopixel algorithm [14], Fig. 1. It is based on curve evolution from seeds placed regularly in the image, which produces a regular “turbopixel” space tessellation. Using various constraints during curve evolution, they encourage a uniform space coverage, compactness of superpixels in the absence of image edges, and boundary alignment when image edges are present. They have to devise a collision detection mechanism to insure no turbopixels overlap. The algorithm runs in seconds on the images in Berkeley dataset [15], as compared to minutes with NC superpixels [1].

We propose a principled approach to compute superpixels in an energy minimization framework. Our method is simple to understand and implement. The basic algorithm, illustrated in Fig. 2, is similar in spirit to texture synthesis [16].



**Fig. 2.** Overview of our algorithm. Left: the original image overlaid with square patches. For clarity, only some patches are shown. Middle: result of patch stitching. Right: superpixel boundaries.

An image is covered with overlapping square patches of fixed size, Fig. 2 left. Each pixel is covered by several patches, and the task is to assign a pixel to one of them. If two neighboring pixels are assigned to the same patch, there is no penalty. If they belong to different patches, then there is a stitching penalty that is inversely proportional to the intensity difference between the pixels. Intuitively, we are stitching patches so that the seams are encouraged to align with intensity edges. The stitching result is in Fig. 2 middle, and superpixel boundaries are in Fig. 2 right. Boundaries are regularized due to the stitching energy function. A superpixel cannot be too large, not larger than a patch size. Small superpixels are discouraged because they contribute a higher cost to the stitching energy. Thus the sizes of superpixels are also regularized. We extend this basic algorithm to other formulations, which allow a trade-off between a less regular space tessellation but more accurate boundaries or better efficiency.

Our work has several advantages over turbopixels [14]. First, we have an explicit energy, and a principled way to optimize it. In contrast, the method in [14] is described only procedurally. Our approach is simpler to understand and analyze. Unlike [14], we do not need a collision detection mechanism, overlap is not allowed by design. Since we have an explicit energy function, we can change its terms to encourage different superpixel types. One modification we add is a term that encourages intensity homogeneity inside a superpixel, not something that is easy to include explicitly into [14]. Another advantage is optimization. Turbopixels are based on level set evolution [17], which is known to have numerical stability issues. We optimize with graph cuts [18], which is known to perform well [19]. Our running time is better. Last, but not least, our approach naturally transfers to 3D for “supervoxel” segmentation of video.

An interesting work on superpixels is in [20,21]. Their goal is somewhat different from ours. They seek superpixels conforming to a grid, which has storage and efficiency advantages. The work in [20] is based on greedy optimization, and [21] uses a more global approach, which, like our work, is also based on graph cuts. However, the formulation in [20,21] poses restrictions on superpixel shapes: the boundary between superpixels cannot “turn back” on itself.

We evaluate our approach on Berkeley dataset [15] and show that we achieve high boundary recall and low undersegmentation error, similar or better than that of [11,14]. We also show that our supervoxels have a high spatial coherence on 3D volumes constructed from video. To show that compact superpixels are more appropriate for some applications, we compare the performance of our superpixels vs. those of [12] on a simple application of salient object segmentation.

## 2 Superpixel Segmentation

In this section we give a detailed description of our superpixel segmentation approach. We review graph cut optimization in Sec. 2.1. Then we explain the basic “compact” superpixel algorithm in Sec. 2.2. In Sec. 2.3 we show how to incorporate variable patch size. The resulting algorithm is called “variable patch” superpixels. Variable patch superpixels are more efficient computationally, and their boundary recall does not suffer a performance loss. However they do have more widely varying sizes. Lastly, in Sec. 2.4 we show how to incorporate intensity constancy constraints, and the resulting algorithm is called “constant intensity” superpixels. Constant intensity superpixels perform better on boundary recall, but, again, have more widely varying sizes.

### 2.1 Energy Minimization with Graph Cuts

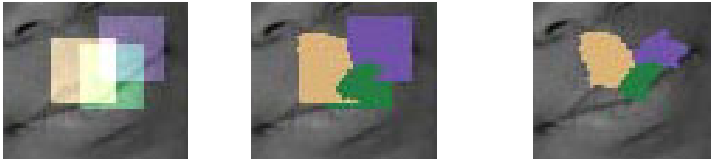
We now briefly review the graph-cut optimization approach [18]. Many problems in vision can be stated as labeling problems. Given a set of pixels  $\mathcal{P}$  and a finite set of labels  $\mathcal{L}$ , the task is to assign a label  $l \in \mathcal{L}$  to each  $p \in \mathcal{P}$ . Let  $f_p$  denote the label assigned to pixel  $p$ , and let  $f$  be the collection of all label assignments. There are two types of constraints. Unary constraints  $D_p(l)$  express how likely is a label  $l$  for pixel  $p$ . Binary constraints  $V_{pq}(l_1, l_2)$  express how likely labels  $l_1$  and  $l_2$  are for neighboring pixels  $p$  and  $q$ . An energy function is:

$$E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \lambda \sum_{\{p,q\} \in \mathcal{N}} w_{pq} \cdot V_{pq}(f_p, f_q), \quad (1)$$

In Eq. (1), the first and the second sums are called the data and the smoothness terms, and  $\mathcal{N}$  is a collection of neighboring pixel pairs. We use 8-connected grid, and Potts model  $V_{pq}(f_p, f_q) = \min(1, |f_p - f_q|)$ . The coefficients  $w_{pq}$  are inversely proportional to the gradient magnitude between  $p$  and  $q$ , encouraging discontinuities to coincide with intensity edges. This energy is NP-hard to optimize. We use the expansion algorithm from [18], which guarantees a factor of 2 approximation. For the max-flow/min-cut algorithm, we use [22].

### 2.2 Compact Superpixels

First recall the intuitive explanation, Fig. 2. We cover an image with overlapping square patches of fixed size, equal to the maximum allowed superpixel size. We



(a) Three patches (b) Their optimal stitching (c) In the final stitching

**Fig. 3.** A simple illustration of patch stitching. Left: orange, green, and purple patches. Middle: their optimal stitching. Right: their optimal stitching in the final result.

seek a stitching of the patches, or, in other words, an assignment of each pixel to a unique patch. The stitches cost cheaper if they coincide with intensity edges. A simple illustration is in Fig. 3. Suppose only three patches in Fig. 3(a) participate. There is a strong intensity gradient on the lip boundary, and therefore the cut between the patches aligns to the lip boundary, Fig. 3(b). Fig. 3(c) shows the shape of these patches in the final stitching, with all patches participating.

We now formalize the problem in the energy minimization framework. We number allowed patches with consecutive integers  $1, \dots, k$ , where  $k$  is the number of allowed patches. We identify  $i$ th patch with an integer label  $i$ , therefore  $\mathcal{L} = \{1, 2, \dots, k\}$ . Even though  $\mathcal{L}$  is ordered, this order has no meaning. Let  $S(l)$  denote the set of the pixels contained in patch  $l \in \mathcal{L}$ . For example, in Fig. 3(a), if  $l$  is the “orange” label, then  $S(l)$  is the set of pixels covered by the orange square. Label  $l$  can be assigned only to pixels in  $S(l)$ . Therefore the data term is:

$$D_p(l) = \begin{cases} 1 & \text{if } p \in S(l) \\ \infty & \text{otherwise} \end{cases} \tag{2}$$

We have to decide how many patches to use and how to spread them out in the image. We address these issues after the energy function is completely specified.

We now discuss the smoothness term. To better approximate Euclidean metric [23] we use 8-connected  $\mathcal{N}$ .  $V_{pq}$  is Potts model with  $w_{pq}$  from [24]:  $w_{pq} = \exp(-\frac{(I_p - I_q)^2}{\text{dist}(p, q) \cdot 2\sigma^2})$ . Here  $I_p$  is the intensity of pixel  $p$ , and  $\text{dist}(p, q)$  is the Euclidean distance between  $p$  and  $q$ .

Observe that with  $D_p$  as defined in Eq. (2), the data term in Eq. (1) is equal for all finite energy labelings. This implies that parameter  $\lambda$  in Eq. (1) has no effect on optimization, so we set  $\lambda = 1$ . Usually  $\lambda$  is an important parameter to choose correctly as it controls the relative weight between the data and the smoothness terms, and, therefore, the length of the boundary. Parameter  $\lambda$  is often set by hand through a tedious trial and error process. In our case the parameter that controls the boundary length is the patch size. Larger patches lead to fewer boundaries in the optimal labeling. Patch size is in some sense a more “natural” parameter since it is chosen by the user to control the size of the maximum superpixel, as appropriate for an application.

We now address the question of how many patches to place in the image. Observe that only some labels (patches) are present in the final labeling. It is

clear from our energy function that the more patches we have, the lower is the final energy, since adding patches only helps to discover better stitches. Thus for the best stitching, we should use a dense strategy, i.e. put a patch at every image pixel. The dense strategy is too expensive. In practice we obtain good results by spreading patches at intervals four times less than the length of the square side.

We use the expansion algorithm [18] for optimization. It does not guarantee an optimum but finds an approximation within a factor of two. We initialize by randomly picking a label  $l$ , and assigning  $l$  to pixels in  $S(l)$  until there are no uninitialized pixels. An intuitive optimization visualization is as follows. An expansion for label  $l$  improves the boundaries under the patch  $S(l)$  and its border.

In addition to the maximum size, the minimum superpixel size is also controlled. Suppose that there is a small superpixel  $A$ . Then for any neighboring superpixel  $B$ , there is no label  $l$  s.t. the patch  $S(l)$  completely covers  $A$  and  $B$ . Otherwise, an expansion on  $l$  would obtain a smaller energy by assigning  $l$  to pixels in  $A \cup B$ , since the boundary between  $B$  and  $A$  disappears and no new boundary is created. The smaller is  $A$ , the less likely it is that there is no neighboring superpixel  $B$  s.t.  $A$  and  $B$  are covered completely by some patch.

Despite a large number of labels, for our energy the expansion algorithm is very efficient. An expansion on label  $l$  needs to be performed only for pixels in  $S(l)$ . This is both memory and time efficient. We run the expansion algorithm for two iterations, and it takes about 5 seconds for Berkeley images [15]. Our algorithm would be easy to implement on multiple processors or GPU.

To summarize, the properties of compact superpixels are as follows. In the presence of large image gradient, superpixel boundaries are encouraged to align with image edges. In the absence of large gradient, superpixels tend to divide space into equally sized regular cells. Superpixel sizes tend to be equalized, and their boundaries are encouraged to be compact by the energy function.

### 2.3 Variable Patch Superpixels

In the previous section we assumed that the patch size is fixed. This helps to ensure that the superpixel sizes are equalized. If one is willing to tolerate a wider variance in superpixel sizes, then it makes sense to allow larger superpixels in the areas with lower image variance.

We develop a simple approach to variable patch superpixels. We allow a variable set of square patches, with the smallest side of size  $k_{min}$  and the largest of size  $k_{max}$ . Let  $S$  be a patch centered at pixel  $p$ . Let  $C(S)$  be the square patch of side twice less than the side of  $S$  also centered at  $p$ , i.e.  $C(S)$  is the “central” part of  $S$ . Let  $P(S)$  be the set of pixels contained in  $S$  but not in  $C(S)$ . As a measure of quality of  $S$  we take  $Q(S) = var(C(S)) - var(P(S))$ . Here  $var(S)$  measures the intensity variance in the patch  $S$ . The lower is  $Q(S)$ , the better is the quality of a patch. That is we want the central part of a patch to be of low variance and the periphery to have a high variance. The expectation is that the inside part of patch  $S$  is not going to contain stitches, and therefore should be uniform in intensity. The cuts are encouraged to lie in the periphery of the patch  $S$ , therefore this part is encouraged to have a high variance.

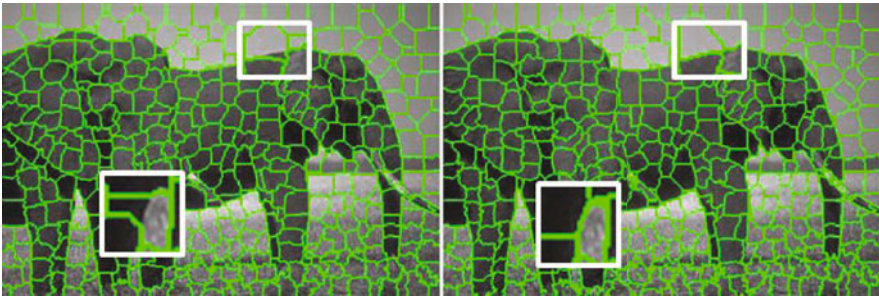
We measure the quality of all possible patches of sizes in the range from  $k_{min}$  to  $k_{max}$ . This can be done efficiently using integral images [10]. After this, we sort all patches in terms of quality and select the  $m$  best ones so that each pixel is contained in at least 4 patches. We found experimentally that variable patch superpixels do not worsen the boundary recall compared to compact superpixels, while improving efficiency by about a factor of 2.

## 2.4 Constant Intensity Superpixels

Since we formulate superpixel segmentation in the energy minimization framework, we can change certain properties of superpixels by simply changing the energy function. We now address one useful change. In the energy for compact superpixels, Sec. 2.2, there is no explicit encouragement that superpixels have constant intensity. Consider a grey and white rectangles adjacent to each other in front of a black background. If there is a patch that covers both rectangles, they will be assigned to the same superpixel, since there is no incentive to split them across two superpixels, regardless of their difference in intensity.

We can explicitly encourage constant intensity inside a superpixel but at the price of obtaining superpixels that are less equalized in terms of size. Let  $c(l)$  be the pixel at the center of patch  $S(l)$ . We change the data term to:

$$D_p(l) = \begin{cases} |I_p - I_{c(l)}| & \text{if } p \in S(l) \\ \infty & \text{otherwise} \end{cases} \quad (3)$$



**Fig. 4.** Two enlarged pieces overlaid over original superpixel images. Left: Compact superpixels, part of the boundaries between elephant legs and on top are missed. Right: constant intensity superpixels, these boundaries are captured.

Now each pixel that is assigned label  $l$  is encouraged to be of the same intensity as the center of patch  $S(l)$ . To ensure this new energy is not increasing during optimization, we have to make sure that that if  $p$  is assigned label  $l$ , then the center of the patch  $c(l)$  is also assigned  $l$ . We can easily do this with addition of the following new term  $T_{new}(f)$  to the energy in Eq. (1):

$$T_{new}(f) = \sum_{p \in \mathcal{P}} W(f_p, f_c(f_p)), \quad (4)$$

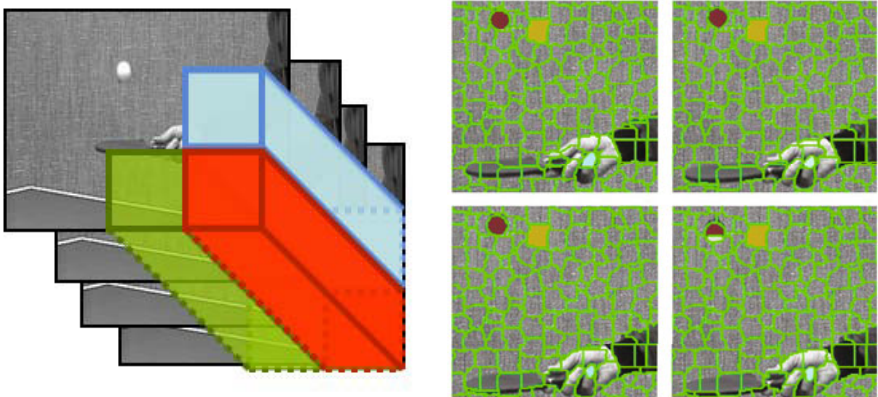
where  $W(\alpha, \beta) = \infty$  if  $\alpha \neq \beta$  and 0 otherwise.

See Fig. 4 for an example where intensity constancy constraint helps to get more accurate boundaries. The cost is that approximately 20% more superpixels are found for the same patch size, some being quite small.

### 3 Supervoxel Segmentation

Our approach naturally extends to segmenting “supervoxels” in 3D space. A voxel has three coordinates  $(x, y, t)$ , with  $t$  being the third dimension. A supervoxel is a set of spatially contiguous voxels that have similar appearance (intensity, color, texture, etc.). Notice that the slices of a voxel at different values of the coordinate  $t$  do not necessarily have the same shape. Segmentation of volumes into supervoxels can be useful, potentially, for medical image and for video processing. In particular, for video processing, there is an interest in coherent 3D segmentation for video abstraction and animation [25,26].

First we create a 3D volume by stacking the frames together, Fig. 5, left. Analogously to the 2D case, we cover the 3D volume by overlapping 3D blocks. For clarity, in Fig. 5 we show only a few non-overlapping blocks. The depth of a block can be different from its width and height. The larger the depth, the more temporal coherency is encouraged. As before, each block corresponds to a label.  $\mathcal{N}$  is now 16-connected and contains neighbors between the frames. Just as in the 2D case, we place blocks overlapping in step size equal to a quarter of the



**Fig. 5.** Supervoxels. Left: video frames are stacked into a 3D volume and covered by a set of 3D blocks. Right: supervoxels, shown separately in each frame. Three supervoxels are highlighted with color (red, yellow, light blue). This figure is better viewed in color.

size of the block (in each dimension). The algorithm is efficient, since we only need to work on a little more than a single block at a time.

Fig. 5, right, shows the results on four consecutive frames of the “tennis” video sequence. We show the section of supervoxels with each frame separately. Notice the high degree of spatial coherency between the frames.

## 4 Experimental Results

First we evaluate how well superpixel boundaries align to image edges. We use Berkeley database [15] that has ground truth provided by human subjects. We use the same measure of boundary recall as in [10,14]. Given a boundary in the ground truth, we search for a boundary in superpixel segmentation within a distance of  $t$  pixels. For experiments we set  $t = 2$ . Recall is the percentage of ground truth boundary that is also present in superpixel segmentation (within a threshold of  $t$ ). Fig. 6(a) plots the dependency of boundary recall on the number of superpixels. The smaller is the number of superpixels, the less boundaries there are, and the worse is the recall. These results were obtained by averaging over 300 images in the database. We compare our compact (OursCompact) and constant intensity (OursIntConstant) superpixels with turbopixels (Turbo), method from [12] (FH), and NC superpixels (NC) [1]. Our variable patch superpixels have performance similar to compact superpixels, so we omit them from Fig. 6 for clarity. From the plot, it is clear that our constant intensity superpixels have a comparable performance to FH and NC methods, and are superior to turbopixels, at least for lower superpixel number. For high superpixel number, all methods have similar performance. Our constant intensity superpixels are superior to compact superpixels for any number of superpixels. The running time of our algorithm is better than that of Turbopixel and NC algorithms.

In Fig. 6(b) shows the undersegmentation error from [14]. Given a ground truth segment and a superpixel segmentation of an image, undersegmentation

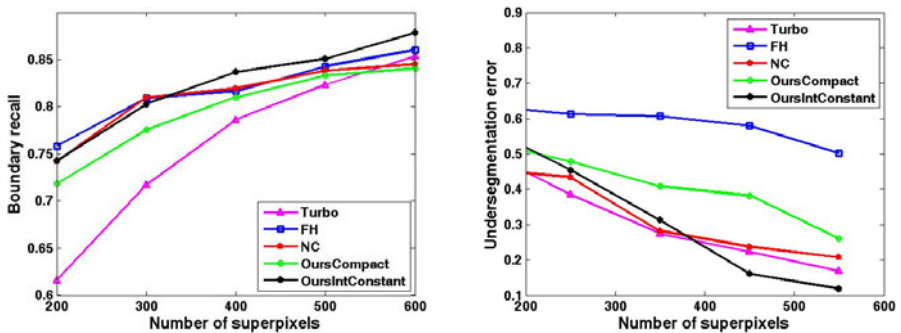
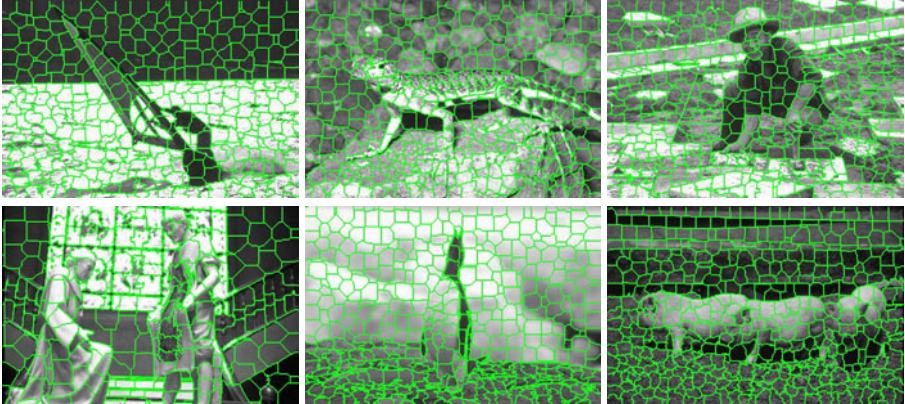


Fig. 6. Performance vs. number of superpixels. Left: boundary recall vs. number of superpixels. Right: undersegmentation error vs. number of superpixels.





**Fig. 7.** Top: compact superpixels, bottom: constant intensity superpixels

error measures what fraction of pixels leak across the boundary of a ground truth segment. The FH algorithm [12] is particularly susceptible to this error because it produces segments of highly variable shapes. Our normalization is slightly different from that in [14], so the vertical axis is on a different scale.

The running times of our algorithms for the images in Berkeley dataset are, on average, as follows. The variable patch superpixels take 2.7 seconds to compute. The compact superpixels take from 5.5 to 7.4 seconds to compute, depending on the patch size. A larger patch size corresponds to a slightly longer running time. The constant intensity superpixels take from 9.7 to 12.3 seconds to compute, again depending on the patch size. Patch sizes are from 20 by 20 to 90 by 90. The turbopixel algorithm [14] takes longer to compute, on average 21.3 seconds. The average running time of NC superpixels [1] is 5.7 minutes.

We now compare the dense strategy of using all patches with the sparse patch placement described in Sec. 2.2. In both cases, we run the expansion algorithm for two iterations. Since the dense strategy is expensive, we ran the experiment for 20 images chosen at random from the Berkeley database [15]. To compare energies across different images, we measure the relative energy difference. For an image  $I$ , let  $E^d(I)$  be the energy with dense patch placement, and  $E^s(I)$  be the energy with the sparse patch placement. Then the relative percent difference in energy is  $100 \cdot \frac{E^s(I) - E^d(I)}{E^d(I)}$ . The mean running time for the dense strategy was 123.5 seconds, whereas for the sparse strategy it was 5.8 seconds. The mean energy difference is 13%, with standard deviation of 0.8%. It makes sense to gain a factor of 21 in computational efficiency while worsening the energy by 12%.

Fig. 1 and Fig. 2 can be used to visually compare our results with turbopixels [14] and NC superpixels [1]. Visually the results are similar, except the NC superpixels appear to have smoother boundaries. This is because in [1] they use a sophisticated boundary detector from [27]. We could incorporate this too in our framework, but it is rather expensive, it takes approximately 30 seconds to

compute boundaries for one image. In Fig. 7 we show some of our segmentations. The top row is compact and the bottom row is intensity constant superpixels.

Fig. 5(b) shows the results on four frames of the “tennis” video sequence. We show the section of supervoxels with each frame separately. Notice the high degree of spatial coherency, even in the areas that are not stationary. Between the first and the last frames shown, the ball moves by about 5 pixels, and the hand by about 8 pixels in the vertical direction. The fingers and the ball are segmented with a high degree of temporal coherency between the frames. We highlight 3 different supervoxels: the one on the ball with red, on the hand with light blue, and on the wall with yellow. The wall is stationary and the supervoxel shape is almost identical between the time slices. The ball and hand are moving, but still the supervoxels slices have a high degree of consistency.

The results of supervoxel segmentation are best to be viewed in a video provided in the supplementary material. We show the original “tennis” sequence and the result of supervoxel segmentation. For visualization, we compute the average intensity of each supervoxel and repaint the video with the average supervoxel intensity. To appreciate the degree of temporal coherence in the supervoxel segmentation, we also perform superpixel segmentation on each frame of the “tennis” sequence separately, using the algorithm in Sec. 2.2. We display the results by painting superpixels with their average intensity. The result of segmentation on each frame separately has much less temporal coherence, as expected. We also provide several other video sequences.

The code for superpixel segmentation will be made available on our web site.

## 5 Application to Salient Object Segmentation

To show that regular superpixels are useful, we evaluated them for salient object segmentation, similar to [28]. The goal is to learn to segment a salient object(s) in an image. We use Berkeley dataset [15], 200 images for training and 100 for testing. Using human marked boundaries as a guide, we manually select salient object(s). Of course, our ground truth is somewhat subjective.

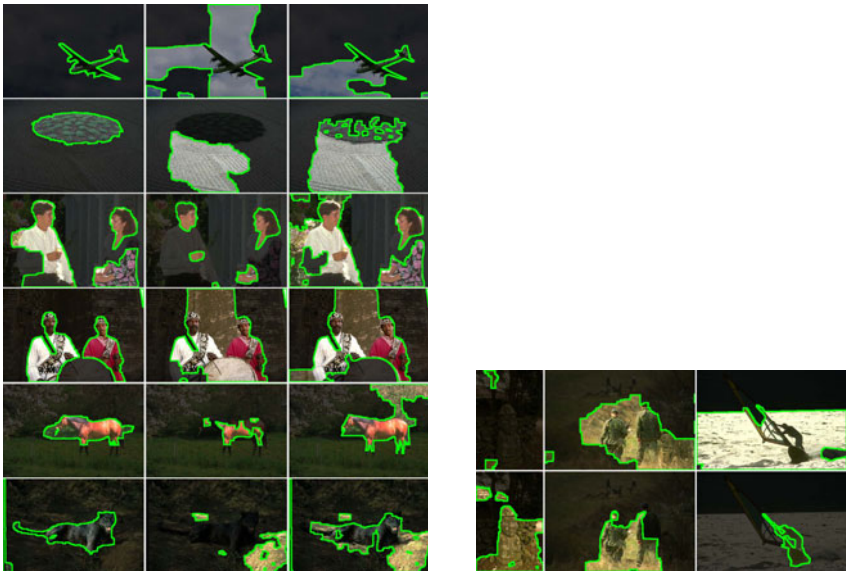
We segment images using rectangular boxes, FH superpixels [12], and our compact superpixels. For boxes, we found that different sizes with overlap give better results. We used 4 different box sizes, from 80 by 80 to 20 by 20. For segmentation, we choose parameters that give the best results on the training data. From each box/superpixel, we extract features similar to those used in [3]. We use features based on color, position (relative to the image size) in the image, and texture. We use gentleboost [29] for training<sup>1</sup>.

The testing error is as follows. Our compact superpixels: 20.5%, FH superpixels [12]: 27.4%, rectangular boxes: 24.0%. Thus performance with our superpixels is significantly better than that of boxes and of FH superpixels [12]. With boxes, the size is controlled, but boxes do not align well to object boundaries. With FH superpixels [12], boundaries are reasonable, but segment size is not controlled,

<sup>1</sup> The implementation by A. Vezhnevets downloaded from [graphics.cs.msu.ru/science/research/machinelearning/adaboosttoolbox](http://graphics.cs.msu.ru/science/research/machinelearning/adaboosttoolbox).

some segments are very large. Thus it appears to be important that our compact superpixels have both regularized size and boundary alignment. We expect that we would have gotten performance similar to ours using turbopixels [14] or NC superpixels [1], but our computational time is much better. Our results in this section are consistent with those of [6], who show that having more accurate spatial support (more accurate superpixels) improves object segmentation.

We also investigate whether the results from classification can be further improved by spatial coherence. We apply the binary segmentation algorithm of [24] to separate an image into the salient object and background components. For the data term, we use the confidences provided by boosting. Using confidences only can smooth results, but will not help to rectify large errors. Additional information is gathered from the histogram of pixels with a high confidence either the object or background class. Thus the data term is computed from quantized color histogram weighted by class confidences. After binary graph cut segmentation the errors are as follows. Our compact superpixels: 21.1%, FH superpixels [12] 25.6%, rectangular boxes 28.4%. Interestingly, the results for FH superpixels [12] improve, results for our superpixels slightly worsen, and results for boxes worsen significantly. Fig. 8(a) shows some results after graph cut segmentation. While what exactly constitutes a salient object may be arguable, our results most often



(a) Left column: results with our superpixels, middle column: results with our superpixels. Bottom row: first result is with FH superpixels [12], last column: results with boxes.

(b) Worst failures. Top row: results with our superpixels. Bottom row: first result is with boxes, the other two are with FH superpixels [12]

**Fig. 8.** Some results for salient object segmentation

correspond to recognizable object(s) occupying a significant portion of a scene, with minimal holes. For most images, results with our superpixels are better or comparable than that of boxes and superpixels of [12]. However sometimes there are significant failures, the worst of them are in Fig. 8(b).

## 6 Future Work

In the future, we plan to investigate more variations on the “basic” energy function to produce superpixels with other interesting properties, such as certain pre-determined orientations, etc. We can also use our algorithm to integrate results from different segmentation algorithms, taking advantages or their respective strengths.

## Acknowledgements

We would like to thank Kyros Kutulakos for bringing the superpixel problem to our attention and for useful discussions. We are grateful to Lena Gorelick for the video sequences used in the supplementary material. This research was supported, in part, by NSERC-DG, NSERC-DAS, CFI, and ERA grants.

## References

1. Ren, X., Malik, J.: Learning a classification model for segmentation. In: ICCV, vol. 1, pp. 10–17 (2003)
2. Mori, G., Ren, X., Efros, A.A., Malik, J.: Recovering human body configurations: combining segmentation and recognition. In: CVPR, vol. 2, pp. 326–333 (2004)
3. Hoiem, D., Efros, A., Hebert, M.: Geometric context from a single image. In: ICCV, pp. 654 – 661 (2005)
4. Mori, G.: Guiding model search using segmentation. In: ICCV, pp. 1417–1423 (2005)
5. He, X., Zemel, R.S., Ray, D.: Learning and incorporating top-down cues in image segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 338–351. Springer, Heidelberg (2006)
6. Malisiewicz, T., Efros, A.A.: Improving spatial support for objects via multiple segmentations. In: BMVC (2007)
7. Pantofaru, C., Schmid, C., Hebert, M.: Object recognition by integrating multiple image segmentations. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 481–494. Springer, Heidelberg (2008)
8. Fulkerson, B., Vedaldi, A., Soatto, S.: Class segmentation and object localization with superpixel neighborhoods. In: ICCV (2009)
9. van den Hengel, A., Dick, A., Thormählen, T., Ward, B., Torr, P.H.S.: Videotrace: rapid interactive scene modelling from video. ACM SIGGRAPH 26, 86 (2007)
10. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR, vol. 1, pp. 511–518 (2001)
11. Comaniciu, D., Meer, P., Member, S.: Mean shift: A robust approach toward feature space analysis. TPAMI 24, 603–619 (2002)
12. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV 59, 167–181 (2004)

13. Shi, J., Malik, J.: Normalized cuts and image segmentation. *TPAMI* 22, 888–905 (1997)
14. Levinshtein, A., Stere, A., Kutulakos, K., Fleet, D., Dickinson, S., Siddiqi, K.: Fast superpixels using geometric flows. *TPAMI* 31, 2290–2297 (2009)
15. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *ICCV*, vol. 2, pp. 416–423 (2001)
16. Kwatra, V., Schödl, A., Essa, I., Turk, G., Bobick, A.: Graphcut textures: Image and video synthesis using graph cuts. *ACM SIGGRAPH* 22, 277–286 (2003)
17. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-Jacobi formulations. *Journal of Computational Physics* 79, 12–49 (1988)
18. Boykov, Y., Veksler, O., Zabih, R.: Efficient approximate energy minimization via graph cuts. *TPAMI* 21, 1222–1239 (2001)
19. 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 with smoothness-based priors. *TPAMI* 30, 1068–1080 (2008)
20. Moore, A., Prince, S., Warrell, J., Mohammed, U., Jones, G.: Superpixel lattices. In: *CVPR* (2008)
21. Moore, A., Prince, S.J., Warrel, J.: Lattice cut - constructing superpixels using layer constraints. In: *CVPR* (2010)
22. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *TPAMI* 24, 137–148 (2004)
23. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: *ICCV*, pp. 26–33 (2003)
24. Boykov, Y., Funka Lea, G.: Graph cuts and efficient n-d image segmentation. *IJCV* 70, 109–131 (2006)
25. Truong, B.T., Venkatesh, S.: Video abstraction: A systematic review and classification. *ACM SIGGRAPH* 3, 3 (2007)
26. Wang, J., Xu, Y., Shum, H., Cohen, M.F.: Video tooning. *ACM SIGGRAPH*, 574–583 (2004)
27. Martin, D., Fowlkes, C., Malik, J.: Learning to find brightness and texture boundaries in natural images. *NIPS* (2002)
28. Liu, T., Sun, J., Zheng, N.N., Tang, X., Shum, H.Y.: Learning to detect a salient object (2007)
29. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: A statistical view of boosting. *The Annals of Statistics* 38, 337–374 (2000)

# Convex Relaxation for Multilabel Problems with Product Label Spaces

Bastian Goldluecke and Daniel Cremers

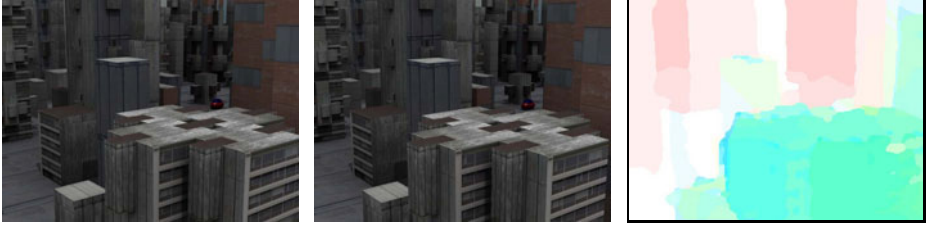
Computer Vision Group, TU Munich

**Abstract.** Convex relaxations for continuous multilabel problems have attracted a lot of interest recently [1,2,3,4,5]. Unfortunately, in previous methods, the runtime and memory requirements scale linearly in the total number of labels, making them very inefficient and often unapplicable for problems with higher dimensional label spaces. In this paper, we propose a reduction technique for the case that the label space is a product space, and introduce proper regularizers. The resulting convex relaxation requires orders of magnitude less memory and computation time than previously, which enables us to apply it to large-scale problems like optic flow, stereo with occlusion detection, and segmentation into a very large number of regions. Despite the drastic gain in performance, we do not arrive at less accurate solutions than the original relaxation. Using the novel method, we can for the first time efficiently compute solutions to the optic flow functional which are within provable bounds of typically 5% of the global optimum.

## 1 Introduction

### 1.1 The Multi-labeling Problem

A multitude of computer vision problems like segmentation, stereo reconstruction and optical flow estimation can be formulated as multi-label problems. In this class of problems, we want to assign to each point  $x$  in an image domain  $\Omega \subset \mathbb{R}^n$  a *label* from a discrete set  $\Gamma = \{1, \dots, N\} \subset \mathbb{N}$ . Assigning the label  $\gamma \in \Gamma$  to  $x$  is associated with the *cost*  $c_\gamma(x) \in \mathbb{R}$ . In computer vision applications, the local costs usually denote how well a given labeling fits some observed data. They can be arbitrarily complex, for instance derived from statistical models or complicated local matching scores. We only assume that the cost functions  $c_\gamma$  lie in the Hilbert space of square integrable functions  $\mathcal{L}^2(\Omega)$ . Aside from the local costs, each possible labeling  $g : \Omega \rightarrow \Gamma$  is penalized by a *regularization term*  $J(g) \in \mathbb{R}$ . The regularizer  $J$  represents our knowledge about which label configurations are a priori more likely. Frequently, it enforces some form of spacial coherence. In this paper, we are above all interested in regularizers which penalize proportionally to the length of the interface between regions with different labels  $\gamma, \chi$  and a metric  $d(\gamma, \chi)$  between the associated labels.



**Fig. 1.** The proposed relaxation method can approximate the solution to multi-labeling problems with a huge number of possible labels by globally solving a convex relaxation model. This example shows two images and the optic flow field between the two, where flow vectors were assigned from a possible set of  $50 \times 50$  vectors, with truncated linear distance as a regularizer. The problem has so many different labels that a solution cannot be computed by alternative relaxation methods on current hardware. See Fig. 7 for the color code of the flow vectors.

The goal is to find a labeling  $g : \Omega \rightarrow \Gamma$  which minimizes the sum of the total costs and the regularizer, i.e.

$$\operatorname{argmin}_{g \in \mathcal{L}^2(\Omega, \Gamma)} J(g) + \int_{\Omega} c_{g(x)}(x) dx. \quad (1)$$

## 1.2 Discrete Approaches

It is well known that in the fully discrete setting, the minimization problem (1) is equivalent to maximizing a Bayesian posterior probability, where the prior probability gives rise to the regularizer [6]. The problem can be stated in the framework of Markov Random Fields [7] and discretized using a graph representation, where the nodes denote discrete pixel locations and the edges encode the energy functional [8].

Fast combinatorial minimization methods based on graph cuts can then be employed to search for a minimizer. In the case that the label space is binary and the regularizer submodular, a global solution of (1) can be found by computing a minimum cut [9,10]. For multi-label problems, one can approximate a solution for example by solving a sequence of binary problems ( $\alpha$ -expansions) [11,12], or linear programming relaxations [13]. Exact solutions to multi-label problems can only be found in some special cases, notably [14], where a cut in a multi-layered graph is computed in polynomial time to find a global optimum. The construction is restricted to convex interaction terms with respect to a linearly ordered label set.

However, in many important scenarios the label space can not be ordered, or a non-convex regularizer is more desirable to better preserve discontinuities in the solution. Even for relatively simple non-convex regularizers like the Potts distance, the resulting combinatorial problem is NP-hard [11]. Furthermore, it is known that the graph-based discretization induces an anisotropy, so that the

solutions suffer from metrication errors [15]. It is therefore interesting to investigate continuous approaches as a possible alternative.

### 1.3 Continuous Approaches

Continuous approaches deal with the multi-label problem by transforming it into a continuous convex problem, obtaining the globally optimal solution, and projecting the continuous solution back onto the original discrete space of labels. Depending on the class of problem, it can be possible to obtain globally optimal solutions to the original discrete minimization problem.

As in the discrete setting, it is possible to solve the two-label problem in a globally optimal way by minimizing a continuous convex energy and subsequent thresholding [2]. In the case of convex interaction terms and a linearly ordered set of labels, there also exists a continuous version of [14] to obtain globally optimal solutions [3]. For the general multi-label case, however, there is no relaxation known which leads to globally optimal solutions of the discrete problem. Currently the most tight relaxation is [4]. The theoretical basis of the reduction technique introduced in this paper is the slightly more transparent formulation introduced in [5] and further generalized in [1], but it can be easily adapted to the framework [4] as well.

The convex relaxation described in [15] works as follows. Instead of looking for  $g$  directly, we associate each label  $\gamma$  with a binary indicator function  $u_\gamma \in \mathcal{L}^2(\Omega, \{0, 1\})$ , where  $u_\gamma(x) = 1$  if and only if  $g(x) = \gamma$ . To make sure that a unique label is assigned to each point, only one of the indicator functions can have the value one. Thus, we restrict optimization to the space

$$U_\Gamma := \left\{ (u_\gamma)_{\gamma \in \Gamma} : u_\gamma \in \mathcal{L}^2(\Omega, \{0, 1\}) \text{ and } \sum_{\gamma \in \Gamma} u_\gamma(x) = 1 \text{ for all } x \in \Omega \right\}. \tag{2}$$

Let  $\langle \cdot, \cdot \rangle$  denote the inner product on the Hilbert space  $\mathcal{L}^2(\Omega)$ , then problem (1) can be written in the equivalent form

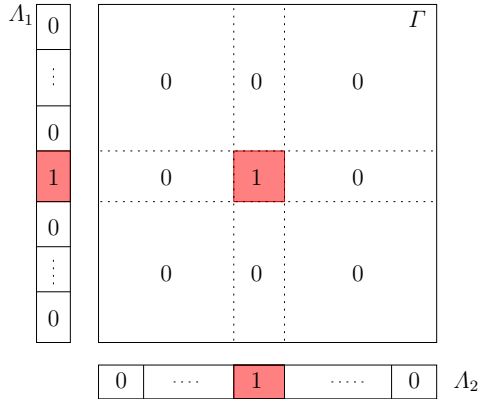
$$\operatorname{argmin}_{\mathbf{u} \in U_\Gamma} J(\mathbf{u}) + \sum_{\gamma \in \Gamma} \langle u_\gamma, c_\gamma \rangle, \tag{3}$$

where we use bold face notation  $\mathbf{u}$  for vectors  $(u_\gamma)_{\gamma \in \Gamma}$  indexed by elements in  $\Gamma$ . We use the same symbol  $J$  to also denote the regularizer on the reduced space. Its definition requires careful consideration, see Section 3.

### 1.4 Contribution: Product Label Spaces

In this work, we discuss label spaces which can be written as a product of a finite number  $d$  of discrete spaces,  $\Gamma = A_1 \times \dots \times A_d$ . Let  $N_j$  be the number of elements in  $A_j$ , then the total number of labels is  $N = N_1 \cdot \dots \cdot N_d$ . In the formulation (3), we optimize over a number of  $N$  binary functions, which can





**Fig. 2.** The central idea of the reduction technique is that if a single indicator function in the product space  $\Gamma$  takes the value 1, then this is equivalent to setting an indicator function in each of the factors  $A_j$ . The memory reduction stems from the fact that there are much more labels in  $\Gamma$  than in all the factors  $A_j$  combined.

be rather large in practical problems. In order to make problems of this form feasible to solve, we present a further reduction which only requires  $N_1 + \dots + N_d$  binary functions - a linear instead of an exponential growth.

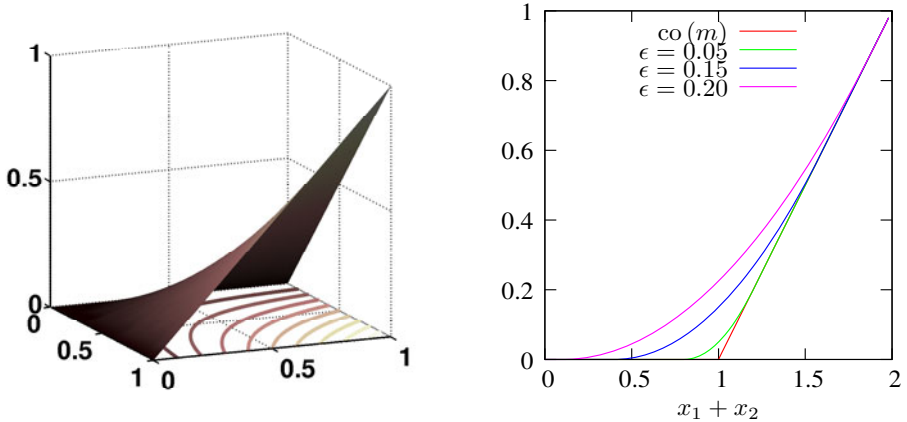
We will show that with our novel reduction technique, it is possible to efficiently solve convex relaxations to multi-label problems which are far too large to approach with previously existing techniques. A prototypical example is optic flow, where a typical total number of labels is around  $32^2$  for practical problems, for which we only require 64 indicator functions instead of 1024. However, the proposed method applies to a much larger class of labelling problems. A consequence of the reduction in variable size is a disproportionately large cut in required runtime, which also makes our method much faster.

## 2 Relaxations for Product Label Spaces

### 2.1 Product Label Spaces

As previously announced, from now on we assume that the space of labels is a product of a finite number  $d$  of discrete spaces,  $\Gamma = A_1 \times \dots \times A_d$ , with  $|A_j| = N_j$ . To each label  $\lambda \in A_j, 1 \leq j \leq d$ , we associate an indicator function  $u_\lambda^j$ . Thus, optimization will take place over the reduced space of functions

$$\begin{aligned}
 \mathcal{U}_\Gamma^\times := & \left\{ (u_\lambda^j)_{1 \leq j \leq d, \lambda \in A_j} : u_\lambda^j \in \mathcal{L}^2(\Omega, \{0, 1\}) \text{ and} \right. \\
 & \left. \sum_{\lambda \in A_j} u_\lambda^j(x) = 1 \text{ for all } x \in \Omega, 1 \leq j \leq d \right\}. \tag{4}
 \end{aligned}$$



(a) Product function  $m(x_1, x_2) = x_1x_2$

(b) Convex envelope  $\text{co}(m)$  and mollified versions for different  $\epsilon$

**Fig. 3.** Product function and its mollified convex envelope for the case  $d = 2$

We use the short notation  $\mathbf{u}^\times$  for a tuple  $(u_\lambda^j)_{1 \leq j \leq d, \lambda \in \Lambda_j}$ . Note that such a tuple consists indeed of exactly  $N_1 + \dots + N_d$  binary functions. The following proposition illuminates the relationship between the function spaces  $\mathcal{U}_\Gamma$  and  $\mathcal{U}_\Gamma^\times$ .

**Proposition 1.** A bijection  $\mathbf{u}^\times \mapsto \mathbf{u}$  from  $\mathcal{U}_\Gamma^\times$  onto  $\mathcal{U}_\Gamma$  is defined by setting

$$u_\gamma := u_{\gamma_1}^1 \cdot \dots \cdot u_{\gamma_d}^d,$$

for all  $\gamma = (\gamma_1, \dots, \gamma_d) \in \Gamma$ .

This is easy to see visually, Figure 2. A formal proof can be found in the appendix. With this new function space, another equivalent formulation to (1) and (3) is

$$\underset{\mathbf{u}^\times \in \mathcal{U}_\Gamma^\times}{\text{argmin}} J(\mathbf{u}^\times) + \sum_{\gamma \in \Gamma} \langle u_{\gamma_1}^1 \cdot \dots \cdot u_{\gamma_d}^d, c_\gamma \rangle. \tag{5}$$

Note that while we have reduced the dimensionality of the problem considerably, we have introduced another difficulty: the data term is not convex anymore, since it contains a product of the components. Thus, in the relaxation, we need to take additional care to make the final problem again convex.

### 2.2 Convex Relaxation

Two steps have to be taken to relax (5) to a convex problem. In a first step, we replace the multiplication function  $m(u_{\gamma_1}^1, \dots, u_{\gamma_d}^d) := u_{\gamma_1}^1 \cdot \dots \cdot u_{\gamma_d}^d$  with a convex function. In order to obtain a tight relaxation, we first move to the convex envelope  $\text{co}(m)$  of  $m$ . Analyzing the epigraph of  $m$ , Fig. 3(a) shows that

$$\text{co}(m)(x_1, \dots, x_d) = \begin{cases} 1 & \text{if } x_1 = \dots = x_d = 1, \\ 0 & \text{if any } x_j = 0. \end{cases} \tag{6}$$

This means that if in the functional,  $m$  is replaced by the convex function  $\text{co}(m)$ , we retain the same binary solutions, as the function values on binary input are the same. We lose nothing on first glance, but on second glance, we forfeited differentiability of the data term, since  $\text{co}(m)$  is not a smooth function anymore.

In order to be able to solve the new problem in practice, we replace  $\text{co}(m)$  again by a mollified function  $\text{co}(m)_\epsilon$ , where  $\epsilon > 0$  is a small constant. We illustrate this for the case  $d = 2$ , where one can easily write down the functions explicitly. In this case, the convex envelope of multiplication is

$$\text{co}(m)(x_1, x_2) = \begin{cases} 0 & \text{if } x_1 + x_2 \leq 1 \\ x_1 + x_2 - 1 & \text{otherwise.} \end{cases}$$

This is a piecewise linear function of the sum of the arguments, i.e symmetric in  $x_1$  and  $x_2$ , see Fig. 3(b). We smoothen the kink by replacing  $\text{co}(m)$  with

$$\text{co}(m)_\epsilon(x_1, x_2) = \begin{cases} 0 & \text{if } x_1 + x_2 \leq 1 - 4\epsilon \\ \frac{1}{16\epsilon}(x_1 + x_2 - (1 - 4\epsilon))^2 & \text{if } 1 - 4\epsilon < x_1 + x_2 < 1 + 4\epsilon \\ 1 & \text{if } x_1 + x_2 \geq 1 + 4\epsilon \end{cases}$$

This function does not satisfy the above condition (6) exactly, but only fulfills the less tight

$$\text{co}(m)_\epsilon(x_1, \dots, x_d) \begin{cases} = 1 & \text{if } x_1 = \dots = x_d = 1, \\ \leq \epsilon & \text{if any } x_j = 0. \end{cases} \tag{7}$$

The following Theorem shows that the solutions of the smoothened energy converge to the solutions of the original energy as  $\epsilon \rightarrow 0$ . After discretization, this means that we obtain an exact solution to the binary problem if we choose  $\epsilon$  small enough, since the problem is combinatorial and the number of possible configurations finite.

**Theorem 1.** *Let  $\epsilon > 0$  and  $\text{co}(m)_\epsilon$  satisfy condition (7). Let  $\mathbf{u}_0^\times$  be a solution to problem (5), and*

$$\mathbf{u}_\epsilon^\times \in \underset{\mathbf{u}^\times \in \mathcal{U}_F^\times}{\text{argmin}} J(\mathbf{u}^\times) + \sum_{\gamma \in \Gamma} \langle \text{co}(m)_\epsilon(u_{\gamma_1}^1, \dots, u_{\gamma_d}^d), c_\gamma \rangle. \tag{8}$$

Then

$$|E_\epsilon(\mathbf{u}_\epsilon^\times) - E(\mathbf{u}_0^\times)| \leq |\Omega| \sum_{\gamma \in \Gamma} \|c_\gamma\|_\infty \epsilon, \tag{9}$$

where  $E$  and  $E_\epsilon$  are the energies of the original problem (5) and smoothened problem (8), respectively.

The proof can be found in the appendix. The key difference of (8) compared to (5) is that the data term is now a convex function.

In the second step of the convex relaxation, we have to make sure the domain of the optimization is a convex set. Thus  $\mathcal{U}_F^\times$  is replaced by its convex

hull  $\text{co}(\mathcal{U}_T^\times)$ . This just means that the domain of the functions  $(u_\lambda^j)$  is extended to the continuous interval  $[0, 1]$ . The final relaxed problem which we are going to solve is now to find

$$\operatorname{argmin}_{\mathbf{u}^\times \in \text{co}(\mathcal{U}_T^\times)} J(\mathbf{u}^\times) + \sum_{\gamma \in \Gamma} \langle \text{co}(m)_\epsilon(u_{\gamma_1}^1, \dots, u_{\gamma_d}^d), c_\gamma \rangle. \tag{10}$$

### 2.3 Numerical Method

With a suitable choice of convex regularizer  $J$ , problem (10) is a continuous convex problem with a convex and differentiable data term. In other relaxation methods, one usually employs fast primal-dual schemes [16,17] to solve the continuous problem. However, those are only applicable to linear data terms. Fortunately, the derivative of the data term is Lipschitz-continuous with Lipschitz constant  $L = \frac{1}{8\epsilon} \sum_\gamma \|c_\gamma\|_2$ . If we take care to choose a lower semi-continuous  $J$ , we are thus in a position to apply the FISTA scheme [18] to the minimization of (10). It is much faster than for example direct gradient descent, with a provable quadratic convergence rate. The remaining problems are how to choose a correct regularizer, and how to get back from a possibly non-binary solution of the relaxed problem to a solution of the original problem.

### 2.4 Obtaining a Solution to the Original Problem

Let  $\hat{\mathbf{u}}^\times$  be a solution to the relaxed problem (10). Thus, the functions  $\hat{u}_\lambda^j$  might have values in between 0 and 1. In order to obtain a feasible solution to the original problem (1), we just project back to the space of allowed functions. The function  $\hat{g} \in \mathcal{L}^2(\Omega, \Gamma)$  closest to  $\hat{\mathbf{u}}^\times$  is given by setting

$$\hat{g}(x) = \operatorname{argmax}_{\gamma \in \Gamma} \hat{u}_{\gamma_1}^1(x) \cdot \dots \cdot \hat{u}_{\gamma_d}^d(x),$$

i.e. we choose the label where the combined indicator functions have the highest value.

We cannot guarantee that the solution  $\hat{g}$  is indeed a global optimum of the original problem (1), since there is nothing equivalent to the thresholding theorem [2] known for this kind of relaxation. However, we still can give a bound how close we are to the global optimum. Indeed, the energy of the optimal solution of (1) must lie somewhere between the energies of  $\hat{\mathbf{u}}^\times$  and  $\hat{g}$ .

## 3 Regularization

The following construction of a family of regularizers is analogous to [1], but extended to accommodate product label spaces. An element  $\mathbf{u}^\times \in \mathcal{U}_T^\times$  can be viewed as a map in  $\mathcal{L}^2(\Omega, \Delta^\times)$ , where

$$\Delta^\times = \Delta^1 \times \dots \times \Delta^d \subset \mathbb{R}^{N_1 + \dots + N_d}$$

and

$$\Delta^i = \left\{ \mathbf{x} \in \{0, 1\}^{N_i} : \sum_{j=1}^{N_i} x_j = 1 \right\}.$$

is the set of corners of the standard  $(k - 1)$ -simplex. As shown previously, there is a one-to-one correspondence between elements in  $\Delta^\times$  and the labels in  $\Gamma$ .

We will now construct a family of regularizers  $J : \text{co}(\mathcal{U}_\Gamma^\times) \rightarrow \mathbb{R}$  and afterwards demonstrate that it is well suited to the problem at hand. For this, we impose a metric  $d$  on the space  $\Gamma$  of labels. A current limitation is that we can only handle the case of separable metrics, i.e.  $d$  must be of the form

$$d(\gamma, \chi) = \sum_{i=1}^d d_i(\gamma_i, \chi_i), \tag{11}$$

where each  $d_i$  is a metric on  $\Delta^i$ . We further assume that each  $d_i$  has an Euclidean representation. This means that each label  $\lambda \in \Delta^i$  shall be *represented* by an  $r_i$ -dimensional vector  $a_\lambda^i \in \mathbb{R}^{r_i}$ , and the distance  $d_i$  defined as Euclidean distance between the representations,

$$d(\lambda, \mu) = |a_\lambda - a_\mu|_2 \text{ for all } \lambda, \mu \in \Delta^i. \tag{12}$$

The goal in the construction of  $J$  is that the higher the distance between labels, the higher shall be the penalty imposed by  $J$ . To make this idea precise, we introduce the linear mappings  $A_i : \text{co}(\Delta^i) \rightarrow \mathbb{R}^{r_i}$  which map labels onto their representations,

$$A_i(\lambda) = a_\lambda^i \text{ for all } \lambda \in \Delta^i.$$

When the labels are enumerated, then in matrix notation, the vectors  $a_\gamma^i$  become exactly the columns of  $A_i$ , which shows the existence of this map.

We can now define the regularizer as

$$J(\mathbf{u}^\times) := \sum_{i=1}^d \text{TV}_v^i(A_i \mathbf{u}^i), \tag{13}$$

where  $\text{TV}_v^i$  is the vectorial total variation on  $\mathcal{L}^2(\Omega, \mathbb{R}^{r_i})$ . The following theorem shows why the above definition makes sense.

**Theorem 2.** *The regularizer  $J$  defined in (13) has the following properties:*

1.  $J$  is convex and positively homogenous on  $\text{co}(\mathcal{U}_\Gamma^\times)$ .
2.  $J(\mathbf{u}^\times) = 0$  for any constant labeling  $\mathbf{u}^\times$ .
3. If  $S \subset \Omega$  has finite perimeter  $\text{Per}(S)$ , then for all labels  $\gamma, \chi \in \Gamma$ ,

$$J(\gamma 1_S + \chi 1_{S^c}) = d(\gamma, \chi) \text{Per}(S),$$

*i.e. a change in labels is penalized proportional to the distance between the labels and the perimeter of the interface.*

The theorem is proved in the appendix. More general classes of metrics on the labels can also be used, see [1]. For the sake of simplicity, we only included the most important example of distances with Euclidean representations. This class includes, but is not limited to, the following special cases:

- The Potts or uniform distance, where  $d_i(\lambda, \mu) = 1$  if and only if  $\lambda = \mu$ , and zero otherwise. This distance function can be achieved by setting  $a_\lambda^i = \frac{1}{2}e_\lambda$ , where  $(e_\lambda)_{\lambda \in \Lambda_i}$  is an orthonormal basis in  $\mathbb{R}^{N_i}$ . All changes between labels are penalized equally.
- The typical case is that the  $a_\lambda^i$  denote feature vectors or actual geometric points, for which  $|\cdot|_2$  is a natural distance. For example, in the case of optic flow, each label corresponds to a flow vector in  $\mathbb{R}^2$ . The representations  $a_\lambda^1, a_\mu^2$  are just real numbers, denoting the possible components of the flow vectors in  $x$  and  $y$ -direction, respectively. The Euclidean distance is a natural distance on the components to regularize the flow field, corresponding to the regularizer of the TV- $L^1$  functional in [19].

The convex functional we wish to minimize is now fully defined, including the regularizer. The ROF type problems with the vectorial total variation as a regularizer, which are at the core of the resulting FISTA scheme, can be minimized with algorithms in [20]. For the also required backprojections onto simplices we recommend the method in [21]. Thus, we can turn our attention towards computing a minimizer in practice. In the remaining section, we will apply the framework to a variety of computer vision problems.

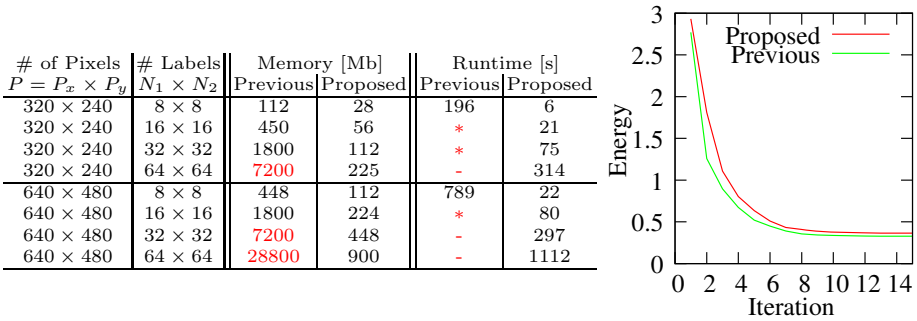
## 4 Experiments

We implemented the proposed algorithm for the case  $d = 2$  on parallel processing GPU architecture using the CUDA programming language, and performed a variety of experiments, with completely different data terms and regularizers. All experiments were performed on an nVidia Tesla C1060 card with 4GB of memory.

When the domain  $\Omega$  is discretized into  $P$  pixels, the primal and dual variables required for the FISTA minimization scheme are represented as matrices. In total, we have to store  $P \cdot (N_1 + \dots + N_d)$  floating point numbers for the primal variables, and  $Pn \cdot (r_1 + \dots + r_d)$  floating point numbers for the dual variables. In contrast, without using our reduction scheme, this number would be as high as  $P \cdot N_1 \cdot \dots \cdot N_d$  for the primal variables and  $Pn \cdot r_1 \cdot \dots \cdot r_d$  for the dual variables, respectively. For the FISTA scheme, we need space for four times the primal variables in total, so we end up with the total values shown in Fig. 4. Thus, problems with large number of labels can only be handled with the proposed reduction technique.

### 4.1 Multi-label Segmentation

For the first example, we chose one with a small label space, so that we can compare the convergence rate and solution energy of the previous method [1]

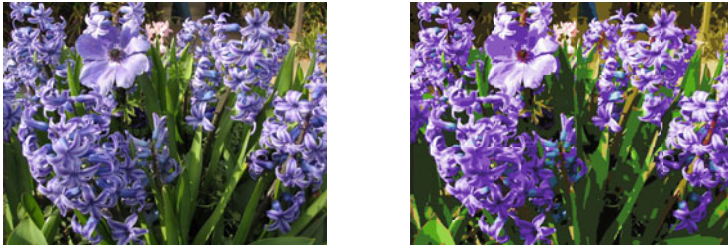


**Fig. 4.** The table shows the total amount of memory required for a FISTA implementation of the previous and proposed methods depending on the size of the problem. Also shown is the total runtime for 15 iterations, which usually suffices for convergence. Numbers shown in red cannot be stored within even the largest of today's CUDA capable cards, so an efficient parallel implementation is not possible. Failures marked with a “\*” are due to another limitation: the shared memory is only sufficient to store the temporary variables for the simplex projection up until dimension 128. In the graph, we see a comparison of the convergence rate between the original scheme and the proposed scheme. Despite requiring significantly less memory and runtime, the relaxation is still sufficiently tight to arrive at an almost similar solution.

with the proposed one. We perform a segmentation of an image based on the HSL color space. The hue and lightness values of the labeling are taken from the discrete sets of equidistant labels  $\mathcal{A}_1$  and  $\mathcal{A}_2$ , respectively. Their size is  $|\mathcal{A}_1| = |\mathcal{A}_2| = 8$ , so there are 64 labels in total, which can still be handled by the old method as well, albeit barely. The labels shall be as close as possible to the original image values, so the cost function penalizes the  $\mathcal{L}^1$ -distance in HSV color space. We choose the regularizer so that the penalty for discontinuities is proportionally larger in regions with higher lightness. The relaxation constant  $\epsilon$  is reduced from 0.2 to 0.05 during the course of the iterations. The result can be seen in Fig. 5, while a comparison of the respective convergence rates are shown in the graph in Fig. 4. The proposed method, despite requiring only a fraction of the memory and computation time, achieves a visually similar result with only a slightly higher energy. Note that the runtime of our method is far lower, since the simplex projection becomes disproportionately more expensive if the length of the vector is increased.

### 4.2 Depth and Occlusion Map

In this test, we simultaneously compute a depth map and an occlusion map for a stereo pair of two color input images  $I_L, I_R : \Omega \rightarrow \mathbb{R}^3$ . The occlusion map shall be a binary map denoting whether a pixel in the left image has a matching pixel in the right image. Thus, the space of labels is two-dimensional with  $\mathcal{A}_1$  consisting of the disparity values and a binary  $\mathcal{A}_2$  for the occlusion map. We use the technique in [1] to approximate a truncated linear smoothness penalty on



**Fig. 5.** Results for the multi-label segmentation. The input image on the left was labelled with  $8 \times 8$  labels in the hue and lightness components of HSL color space. The label distance is set so that smoothing is stronger in darker regions, which creates an interesting visual effect.



**Fig. 6.** The proposed method can be employed to simultaneously optimize for a displacement and an occlusion map. This problem is also too large to be solved by alternative relaxation methods on current GPUs. From left to right: (a) Left input image  $I_L$ . (b) Right input image  $I_R$ . (c) Computed disparity and occlusion map, red areas denote occluded pixels.

the disparity values. A Potts regularizer is imposed for the occlusion map. The distance on the label space thus becomes

$$d(\gamma, \chi) = s_1 \min(t_1, |\gamma_1 - \chi_1|) + s_2 |\gamma_2 - \chi_2|, \tag{14}$$

with suitable weights  $s_1, s_2 > 0$  and threshold  $t_1 > 0$ . We penalize an occluded pixel with a constant cost  $c_{occ} > 0$ , which corresponds to a threshold for the similarity measure above which we believe that a pixel is not matched correctly anymore. The cost associated with a label  $\gamma$  at  $(x, y) \in \Omega$  is then defined as

$$c_\gamma(x, y) = \begin{cases} c_{occ} & \text{if } \gamma_2 = 1, \\ \|I_L(x, y) - I_R(x - \lambda_1, y)\|_2 & \text{otherwise.} \end{cases} \tag{15}$$

The result for the “Moebius” test pair from the Middlebury benchmark is shown in Fig. 6. The input image resolution was scaled to  $640 \times 512$ , requiring 128 disparity labels, which resulted in a total memory consumption which was slightly too big for previous methods, but still in reach of the proposed algorithm. Total computation time required was 597 seconds.





**Fig. 7.** When employed for optic flow, the proposed method can successfully capture large displacements without the need for coarse-to-fine approaches, since a global optimization is performed over all labels. In contrast to existing methods, our solution is within a known bound of the global optimum.

### 4.3 Optic Flow

In the final test, we compute optic flow between two color input images  $I_0, I_1 : \Omega \rightarrow \mathbb{R}^3$  taken at two different time instants. The space of labels is again two-dimensional, with  $A_1 = A_2$  denoting the possible components of flow vectors in  $x$  and  $y$ -direction, respectively. We regularize both directions with a truncated linear penalty on the component distance, i.e.

$$d(\gamma, \chi) = s \min(t, |\gamma_1 - \chi_1|) + s \min(t, |\gamma_2 - \chi_2|), \quad (16)$$

with a suitable weight  $s > 0$  and threshold  $t > 0$ . The cost function just compares pointwise pixel colors in the images, i.e.

$$c_\gamma(x, y) = \|I_0(x, y) - I_1(x + \gamma_1, y + \gamma_2)\|_2. \quad (17)$$

Results can be observed in Fig. 11 and 7. Due to the global optimization of a convex energy, we can successfully capture large displacements without having to implement a coarse-to-fine scheme. The number of labels is  $50 \times 50$  at an image resolution of  $640 \times 480$ , so the memory requirements are so high that this problem is currently impossible to solve with previous convex relaxation techniques by a large margin, see Fig. 4. Total computation time using our method was 678 seconds. A comparison of the energies of the continuous and discretized solution shows that we are within 5% of the global optimum for all examples.

## 5 Conclusion

We have introduced a continuous convex relaxation for multi-label problems where the label space is a product space. Such labeling problems are plentiful in computer vision. The proposed reduction method improves on previous methods in that it requires orders of magnitude less memory and computation time, while retaining the advantages: a very flexible choice of distance on the label space, a globally optimal solution of the relaxed problem and an efficient parallel GPU implementation with guaranteed convergence.

Because of the reduced memory requirements, we can successfully handle specific problems with very large number of labels, which could not be attempted with previous convex relaxation techniques. Among other examples we presented a convex relaxation for the optic flow functional with truncated linear penalizer on the distance between the flow vectors. To our knowledge, this is the first relaxation for this functional which can be optimized globally and efficiently.

## References

1. Lellmann, J., Becker, F., Schnörr, C.: Convex optimization for multi-class image labeling with a novel family of total variation based regularizers. In: IEEE International Conference on Computer Vision (ICCV) (2009)
2. Nikolova, M., Esedoglu, S., Chan, T.: Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal of Applied Mathematics* 66, 1632–1648 (2006)
3. Pock, T., Schoenemann, T., Graber, G., Bischof, H., Cremers, D.: A convex formulation of continuous multi-label problems. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 792–805. Springer, Heidelberg (2008)
4. Pock, T., Chambolle, A., Bischof, H., Cremers, D.: A convex relaxation approach for computing minimal partitions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 810–817 (2009)
5. Zach, C., Gallup, D., Frahm, J., Niethammer, M.: Fast global labeling for real-time stereo using multiple plane sweeps. In: *Vision, Modeling and Visualization*, pp. 243–252 (2009)
6. Szeliski, R.: Bayesian modeling of uncertainty in low-level vision. *International Journal of Computer Vision* 5, 271–301 (1990)
7. Kindermann, R., Snell, J.: *Markov Random Fields and Their Applications*. American Mathematical Society, Providence (1980)
8. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: IEEE International Conference on Computer Vision (ICCV), pp. 26–33 (2003)
9. Greig, D., Porteous, B., Seheult, A.: Exact maximum a posteriori estimation for binary images. *J. Royal Statistics Soc.* 51, 271–279 (1989)
10. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 147–159 (2004)
11. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* 23, 1222–1239 (2001)
12. Schlesinger, D., Flach, B.: Transforming an arbitrary min-sum problem into a binary one. Technical report, Dresden University of Technology (2006)
13. Wainwright, M., Jaakkola, T., Willsky, A.: Map estimation via agreement on trees: message-passing and linear programming. *IEEE Trans. Inf. Theory* 51, 3697–3717 (2005)
14. Ishikawa, H.: Exact optimization for markov random fields with convex priors. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 1333–1336 (2003)
15. Klodt, M., Schoenemann, T., Kolev, K., Schikora, M., Cremers, D.: An experimental comparison of discrete and continuous shape optimization methods. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 332–345. Springer, Heidelberg (2008)
16. Popov, L.: A modification of the arrow-hurwicz method for search of saddle points. *Math. Notes* 28, 845–848 (1980)

17. Nesterov, Y.: Smooth minimization of non-smooth functions. *Math. Prog.* 103, 127–152 (2004)
18. Beck, A., Teboulle, M.: Fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences* 2, 183–202 (2009)
19. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime  $TV - L^1$  optical flow. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) *DAGM 2007. LNCS*, vol. 4713, pp. 214–223. Springer, Heidelberg (2007)
20. Duval, V., Aujol, J.F., Vese, L.: Projected gradient based color image decomposition. In: Tai, X.-C., Mørken, K., Lysaker, M., Lie, K.-A. (eds.) *Scale Space and Variational Methods in Computer Vision. LNCS*, vol. 5567, pp. 295–306. Springer, Heidelberg (2009)
21. Michelot, C.: A finite algorithm for finding the projection of a point onto the canonical simplex of  $\mathbb{R}^n$ . *J. Optimization Theory and Applications* 50, 195–200 (1986)

## Appendix

**Proof of Proposition 1.** In order to proof the proposition, we have to show that the mapping induces a point-wise bijection from  $\Delta^\times$  onto

$$\Delta = \left\{ \mathbf{x} \in \{0, 1\}^N : \sum_{j=1}^N x_j = 1 \right\}.$$

We first show it is onto: for  $\mathbf{u}(x)$  in  $\Delta$ , there exists exactly one  $\gamma \in \Gamma$  with  $u_\gamma(x) = 1$ . Set  $u_\lambda^i(x) = 1$  if  $\lambda = \gamma_i$ , and  $u_\lambda^i(x) = 0$  otherwise. Then  $\mathbf{u}(x) = \mathbf{u}^1(x) \cdot \dots \cdot \mathbf{u}^d(x)$ , as desired. To see that the map is one-to-one, we just count the elements in  $\Delta^\times$ . Since  $\Delta^i$  contains  $N_i$  elements, the number of elements in  $\Delta^\times$  is  $N_1 \cdot \dots \cdot N_d = N$ , the same as in  $\Delta$ .  $\square$

**Proof of Theorem 1.** The regularizers of the original and smoothed problems are the same, so because of condition (7),

$$|E_\epsilon(\mathbf{u}_\epsilon^\times) - E(\mathbf{u}_0^\times)| \leq \left| \sum_{\gamma \in \Gamma} \int_{\Omega} \epsilon c_\gamma \, dx \right| \leq |\Omega| \sum_{\gamma \in \Gamma} \|c_\gamma\|_\infty \epsilon. \tag{18}$$

This completes the proof.  $\square$

**Proof of Theorem 2.** The first two claims are basic properties of the total variation. For the last claim, we combine Corollary 1 in [1] with the definition of the metric in equations (11) and (12) to find

$$\begin{aligned} J(\gamma 1_S + \chi 1_{S^c}) &= \sum_{i=1}^d \text{TV}_v^i(A_i(\gamma 1_S + \chi 1_{S^c})) = \sum_{i=1}^d |a_\gamma^i - a_\chi^i|_2 \text{Per}(S) \\ &= d(\gamma, \chi) \text{Per}(S). \end{aligned} \tag{19}$$

This completes the proof.  $\square$

# Graph Cut Based Inference with Co-occurrence Statistics<sup>\*</sup>

Lubor Ladicky<sup>1,\*\*</sup>, Chris Russell<sup>1,\*\*</sup>, Pushmeet Kohli<sup>2</sup>, and Philip H.S. Torr<sup>1</sup>

<sup>1</sup> Oxford Brookes

<sup>2</sup> Microsoft Research

**Abstract.** Markov and Conditional random fields (CRFs) used in computer vision typically model only local interactions between variables, as this is computationally tractable. In this paper we consider a class of global potentials defined over all variables in the CRF. We show how they can be readily optimised using standard graph cut algorithms at little extra expense compared to a standard pairwise field.

This result can be directly used for the problem of *class based image segmentation* which has seen increasing recent interest within computer vision. Here the aim is to assign a label to each pixel of a given image from a set of possible object classes. Typically these methods use random fields to model local interactions between pixels or super-pixels. One of the cues that helps recognition is global *object co-occurrence statistics*, a measure of which classes (such as chair or motorbike) are likely to occur in the same image together. There have been several approaches proposed to exploit this property, but all of them suffer from different limitations and typically carry a high computational cost, preventing their application on large images. We find that the new model we propose produces an improvement in the labelling compared to just using a pairwise model.

## 1 Introduction

Class based image segmentation is a highly active area of computer vision research as is shown by a spate of recent publications [11][22][29][31][34]. In this problem, every pixel of the image is assigned a choice of object class label, such as grass, person, or dining table. Formulating this problem as a likelihood, in order to perform inference, is a difficult problem, as the cost or energy associated with any labelling of the image should take into account a variety of cues at different scales. A good labelling should take account of: low-level cues such as colour or texture [29], that govern the labelling of single pixels; mid-level cues such as region continuity, symmetry [23] or shape [2] that govern the assignment of regions within the image; and high-level statistics that encode inter-object relationships, such as which objects can occur together in a scene. This combination of cues makes for a multi-scale cost function that is difficult to optimise.

Current state of the art low-level approaches typically follow the methodology proposed in *Texton-boost* [29], in which weakly predictive features such as colour, location,

---

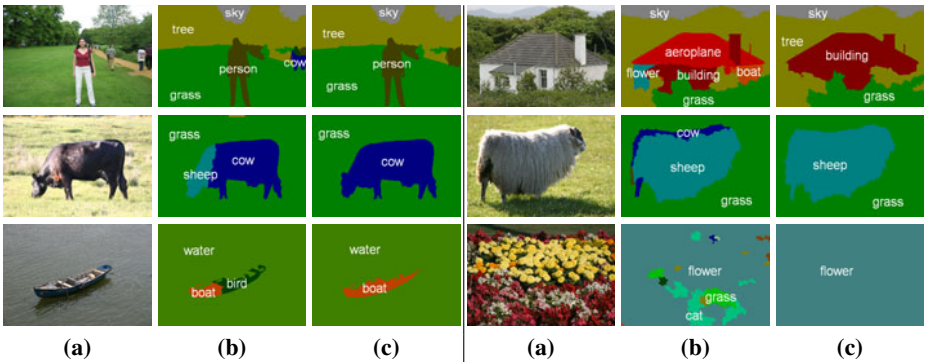
<sup>\*</sup> This work was supported by EPSRC, HMGCC and the PASCAL2 Network of Excellence. Professor Torr is in receipt of a Royal Society Wolfson Research Merit Award.

<sup>\*\*</sup> The authors assert equal contribution and joint first authorship.

and texon response are used to learn a classifier which provides costs for a single pixel taking a particular label. These costs are combined in a contrast sensitive Conditional Random Field CRF [19].

The majority of mid-level inference schemes [25,20] do not consider pixels directly, rather they assume that the image has been segmented into super-pixels [5,8,28]. A labelling problem is then defined over the set of regions. A significant disadvantage of such approaches is that mistakes in the initial over-segmentation, in which regions span multiple object classes, cannot be recovered from. To overcome this [10] proposed a method of reshaping super-pixels to recover from the errors, while the work [17] proposed a novel hierarchical framework which allowed for the integration of multiple region-based CRFs with a low-level pixel based CRF, and the elimination of inconsistent regions.

These approaches can be improved by the inclusion of costs based on high level statistics, including object class co-occurrence, which capture knowledge of scene semantics that humans often take for granted: for example the knowledge that cows and sheep are not kept together and less likely to appear in the same image; or that motorbikes are unlikely to occur near televisions. In this paper we consider object class co-occurrence to be a measure of how likely it is for a given set of object classes to occur together in an image. They can also be used to encode scene specific information such as the facts that computer monitors and stationary are more likely to occur in offices, or that trees and grass occur outside. The use of such costs can help prevent some of the most glaring failures in object class segmentation, such as the labelling of a cow as half cow and half sheep, or the mistaken labelling of a boat surrounded by water as a book.



**Fig. 1. Best viewed in colour:** Qualitative results of object co-occurrence statistics. (a) Typical images taken from the MSRC data set [29]; (b) A labelling based upon a pixel based random field model [17] that does not take into account co-occurrence; (c) A labelling of the same model using co-occurrence statistics. The use of co-occurrence statistics to guide the segmentation results in a labelling that is more parsimonious and more likely to be correct. These co-occurrence statistics suppress the appearance of small unexpected classes in the labelling. **Top left:** a mistaken hypothesis of a cow is suppressed **Top right:** Many small classes are suppressed in the image of a building. Note that the use of co-occurrence typically changes labels, but does not alter silhouettes.

As well as penalising strange combinations of objects appearing in an image, co-occurrence potentials can also be used to impose an MDL<sup>1</sup> prior that encourages a parsimonious description of an image using fewer labels. As discussed eloquently in the recent work [4], the need for a bias towards parsimony becomes increasingly important as the number of classes to be considered increases.

Figure 1 illustrates the importance of co-occurrence statistics in image labelling.

The promise of co-occurrence statistics has not been ignored by the vision community. In [22] Rabinovich *et al.* proposed the integration of such co-occurrence costs that characterise the relationship between two classes. Similarly Torralba *et al.* [31] proposed scene-based costs that penalised the existence of particular classes in a context dependent manner. We shall discuss these approaches, and some problems with them in the next section.

## 2 CRFs and Co-occurrence

A conventional CRF is defined over a set of random variables  $\mathcal{V} = \{1, 2, 3, \dots, n\}$  where each variable takes a value from the label set  $\mathcal{L} = \{l_1, l_2, \dots, l_k\}$  corresponding to the set of object classes. An assignment of labels to the set of random variables will be referred to as a *labelling*, and denoted as  $\mathbf{x} \in \mathcal{L}^{|\mathcal{V}|}$ . We define a cost function  $E(\mathbf{x})$  over the CRF of the form:

$$E(\mathbf{x}) = \sum_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c) \quad (1)$$

where the potential  $\psi_c$  is a cost function defined over a set of variables (called a clique)  $c$ , and  $\mathbf{x}_c$  is the state of the set of random variables that lie within  $c$ . The set  $\mathcal{C}$  of cliques is a subset of the power set of  $\mathcal{V}$ , *i.e.*  $\mathcal{C} \subseteq P(\mathcal{V})$ . In the majority of vision problems, the potentials are defined over a clique of size at most 2. *Unary potentials* are defined over a clique of size one, and typically based upon classifier responses (such as ada-boost [29] or kernel SVMs [27]), while *pairwise potentials* are defined over cliques of size two and model the correlation between pairs of random variables.

### 2.1 Incorporating Co-occurrence Potentials

To model object class co-occurrence statistics a new term  $K(\mathbf{x})$  is added to the energy:

$$E(\mathbf{x}) = \sum \psi_c(\mathbf{x}_c) + K(\mathbf{x}). \quad (2)$$

The question naturally arises as to what form an energy involving co-occurrence terms should take. We now list a set of desiderata that we believe are intuitive for any co-occurrence cost.

(i) *Global Energy*: We would like a formulation of co-occurrence that allows us to estimate the segmentation using all the data directly, by minimising a *single* cost function of the form (2). Rather than any sort of two stage process in which a hard decision is made of which objects are present in the scene *a priori* as in [31].

<sup>1</sup> Minimum description length.

(ii) *Invariance*: The co-occurrence cost should depend only on the labels present in an image, it should be invariant to the number and location of pixels that object occupies. To reuse an example from [32], the surprise at seeing a polar bear in a street scene should not vary with the number of pixels that represent the bear in the image.

(iii) *Efficiency*: Inference should be tractable, *i.e.* the use of co-occurrence should not be the bottle-neck preventing inference. As the memory requirements of any conventional inference algorithm [30] is typically  $O(|\mathcal{V}|)$  for vision problems, the memory requirements of a formulation incorporating co-occurrence potentials should also be  $O(|\mathcal{V}|)$ .

(iv) *Parsimony*: The cost should follow the principle of parsimony in the following way: if several solutions are almost equally likely then the solution that can describe the image using the fewest distinct labels should be chosen. Whilst this might not seem important when classifying pixels into a few classes, as the set of putative labels for an image increases the chance of speckle noise due to misclassification will increase unless a parsimonious solution is encouraged.

While these properties seem uncontroversial, no prior work exhibits property (ii). Similarly, no approaches satisfy properties (i) and (iii) simultaneously. In order to satisfy condition (ii) the co-occurrence cost  $K(\mathbf{x})$  defined over  $\mathbf{x}$  must be a function defined on the set of labels  $L(\mathbf{x}) = \{l \in \mathcal{L} : \exists x_i = l\}$  present in the labelling  $\mathbf{x}$ ; this guarantees invariance to the size of an object:

$$K(\mathbf{x}) = C(L(\mathbf{x})) \quad (3)$$

Embedding the co-occurrence term in the CRF cost function (II), we have:

$$E(\mathbf{x}) = \sum_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c) + C(L(\mathbf{x})). \quad (4)$$

To satisfy the parsimony condition (iv) potentials must act to penalise the unexpected appearance of combinations of labels in a labelling. This observation can be formalised as the statement that the cost  $C(L)$  monotonically increasing with respect to the label set  $L$  *i.e.* :

$$L_1 \subset L_2 \implies C(L_1) \leq C(L_2). \quad (5)$$

The new potential  $C(L(\mathbf{x}))$  can be seen as a particular higher order potential defined over a clique which includes the whole of  $\mathcal{V}$ , *i.e.*  $\psi_{\mathcal{V}}(\mathbf{x})$ .

## 2.2 Prior Work

There are two existing approaches to co-occurrence potentials, neither of which uses potentials defined over a clique of size greater than two. The first makes an initial hard estimate of the type of scene, and updates the unary potentials associated with each pixel to encourage or discourage particular choices of label, on the basis of how likely they are to occur in the scene. The second approach models object co-occurrence as a pairwise potential between regions of the image.

Torralba *et al.* [31] proposed the use of additional unary potentials to capture scene based occurrence priors. Their costs took the form:

$$K(\mathbf{x}) = \sum_{i \in \mathcal{V}} \phi(x_i). \quad (6)$$

While the complexity of inference over such potentials scales linearly with the size of the graph, they are prone to over counting costs, violating (ii), and require an initial hard decision of scene type before inference, which violates (i). As it encourages the appearance of all labels which are common to a scene, it does not necessarily encourage parsimony (iv).

A similar approach was seen in the Pascal VOC2008 object segmentation challenge, where the best performing method, by Csurka [6], worked in two stages. Initially the set of object labels present in the image was estimated, and in the second stage, a label from the estimated label set was assigned to each image pixel. As no cost function  $K(\cdot)$  was proposed, it is open to debate if it satisfied (ii) or (iv).

Method	Global energy	Invariance	Efficiency	Parsimony
	(i)	(ii)	(iii)	(iv)
Unary [31]	✗	✗	✓	✗
Pairwise [22,9,32]	✓	✗	✗	✓
Csurka [6]	✗	—	✓	—
Our approach	✓	✓	✓	✓

**Fig. 2.** A comparison of the capabilities of existing image co-occurrence formulations against our new approach. See section 2.2 for details.

Rabinovich *et al.* [9,22], and independently [32], proposed co-occurrence as a soft constraint that approximated  $C(L(\mathbf{x}))$  as a pairwise cost defined over a *fully connected graph* that took the form:

$$K(\mathbf{x}) = \sum_{i,j \in \mathcal{V}} \phi(x_i, x_j), \quad (7)$$

where  $\phi$  was some potential which penalised labels that should not occur together in an image. Unlike our model (4) the penalty cost for the presence of pairs of labels, that rarely occur together, appearing in the same image grows with the *number* of random variables taking these labels, violating assumption (ii). While this serves as a functional penalty that prevents the occurrence of many classes in the same labelling, it does not accurately model the co-occurrence costs we described earlier. The memory requirements of inference scales badly with the size of a fully connected graph. It grows with complexity  $O(|\mathcal{V}|^2)$  rather than  $O(|\mathcal{V}|)$  with the size of the graph, violating constraint (iii). Providing the pairwise potentials are semi-metric [3], it does satisfy the parsimony condition (iv).

To minimise these difficulties, previous approaches defined variables over segments rather than pixels. Such segment based methods work under the assumption that some segments share boundaries with objects in the image. This is not always the case, and this assumption may result in dramatic errors in the labelling. The relationship between previous approaches and the desiderata can be seen in figure 2.



Two efficient schemes [7][12] have been proposed for the minimisation of the number of classes or objects present in a scene. While neither of them directly models class based co-occurrence relationships, their optimisation approaches do satisfy our desiderata.

One such approach was proposed by Hoiem *et al.* [12], who used a cost based on the number of objects in the scene, in which the presence of any instance of any object incurs a uniform penalty cost. For example, the presence of both a motorbike and a bus in a single image is penalised as much as the presence of two buses. Minimising the number of objects in a scene is a good method of encouraging consistent labellings, but does not capture any co-occurrence relationship between object classes.

In a recent work, appearing at the same time as ours, DeLong *et al.* [7] proposed the use of a soft cost over the number of labels present in an image for clustering. While the mathematical formulation they propose is more flexible than this, they do not suggest any applications of this increased flexibility. Moreover, their formulation is less general than ours as it does not support the full range of monotonically increasing label set costs.

### 3 Inference on Global Co-occurrence Potentials

Consider the energy (4) defined in section 2.1. The inference problem becomes:

$$\begin{aligned} \mathbf{x}^* &= \arg \min_{\mathbf{x} \in \mathcal{L}^{|\mathcal{V}|}} \sum_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c) + C(L(\mathbf{x})) \\ \text{s.t. } \mathbf{x} &\in \mathcal{L}^{|\mathcal{V}|}, L(\mathbf{x}) = \{l \in \mathcal{L} : \exists x_i = l\}. \end{aligned} \quad (8)$$

In the general case the problem of minimising this energy can be reformulated as an integer program and approximately solved as an LP-relaxation [16]. This LP-formulation can be transformed using a Lagrangian relaxation into a pairwise energy, allowing algorithms, such as Belief Propagation [33] or TRW-S [14], that can minimise arbitrary pairwise energies to be applied [16]. However, reparameterisation methods such as these perform badly on densely connected graphs [15][26].

In this section we show that under assumption, that  $C(L)$  is monotonically increasing with respect to  $L$ , the problem can be solved efficiently using  $\alpha\beta$ -swap and  $\alpha$ -expansion moves [3], where the number of additional edges of the graph grows linearly with the number of variables in the graph. In contrast to [22], these algorithms can be applied to large graphs with more than 200,000 variables.

Move making algorithms project the problem into a smaller subspace in which a sub-problem is efficiently solvable. Solving this sub-problem proposes optimal moves which guarantee that the energy decreases after each move and must eventually converge. The performance of move making algorithms depends dramatically on the size of the move space. The expansion and swap move algorithms we consider project the problem into two label sub-problem and under the assumption that the projected energy is pairwise and submodular, it can be solved using graph cuts. Because the energy (4) is additive, we derive graph constructions only for term  $C(L(\mathbf{x}))$ . Both the application of swap and expansion moves to minimise the energy, and the graph construction for the other terms proceed as described in [3].

### 3.1 $\alpha\beta$ -Swap Moves

The swap and expansion move algorithms can be encoded as a vector of binary variables  $\mathbf{t} = \{t_i, \forall i \in \mathcal{V}\}$ . The transformation function  $T(\mathbf{x}^p, \mathbf{t})$  of a move algorithm takes the current labelling  $\mathbf{x}^p$  and a move  $\mathbf{t}$  and returns the new labelling  $\mathbf{x}$  which has been induced by the move.

In an  $\alpha\beta$ -swap move every random variable  $x_i$  whose current label is  $\alpha$  or  $\beta$  can transition to a new label of  $\alpha$  or  $\beta$ . One iteration of the algorithm involves making moves for all pairs  $(\alpha, \beta)$  in  $\mathcal{L}^2$  successively. The transformation function  $T_{\alpha\beta}(x_i, t_i)$  for an  $\alpha\beta$ -swap transforms the label of a random variable  $x_i$  as:

$$T_{\alpha\beta}(x_i, t_i) = \begin{cases} \alpha & \text{if } x_i = \alpha \text{ or } \beta \text{ and } t_i = 0, \\ \beta & \text{if } x_i = \alpha \text{ or } \beta \text{ and } t_i = 1. \end{cases} \quad (9)$$

Consider a swap move over the labels  $\alpha$  and  $\beta$ , starting from an initial label set  $L(\mathbf{x})$ . We assume that either  $\alpha$  or  $\beta$  is present in the image. Then, after a swap move the labels present must be an element of  $S$  which we define as:

$$S = \{L(\mathbf{x}) \cup \{\alpha\} \setminus \{\beta\}, L(\mathbf{x}) \cup \{\beta\} \setminus \{\alpha\}, L(\mathbf{x}) \cup \{\alpha, \beta\}\}. \quad (10)$$

Let  $\mathcal{V}_{\alpha\beta}$  be the set of variables currently taking label  $\alpha$  or  $\beta$ . The move energy for  $C(L(\mathbf{x}))$  is:

$$E(\mathbf{t}) = \begin{cases} C_\alpha = C(L(\mathbf{x}) \cup \{\alpha\} \setminus \{\beta\}) & \text{if } \forall i \in \mathcal{V}_{\alpha\beta}, t_i = 0, \\ C_\beta = C(L(\mathbf{x}) \cup \{\beta\} \setminus \{\alpha\}) & \text{if } \forall i \in \mathcal{V}_{\alpha\beta}, t_i = 1, \\ C_{\alpha\beta} = C(L(\mathbf{x}) \cup \{\alpha, \beta\}) & \text{otherwise.} \end{cases} \quad (11)$$

Note that, if  $C(L)$  is monotonically increasing with respect to  $L$  then, by definition,  $C_\alpha \leq C_{\alpha\beta}$  and  $C_\beta \leq C_{\alpha\beta}$ .

**Lemma 1.** For a function  $C(L)$ , monotonically increasing with respect to  $L$ , the move energy can be represented as a binary submodular pairwise cost with two auxiliary variables  $z_\alpha$  and  $z_\beta$  as:

$$E(\mathbf{t}) = C_\alpha + C_\beta - C_{\alpha\beta} + \min_{z_\alpha, z_\beta} \left[ (C_{\alpha\beta} - C_\alpha)z_\beta + (C_{\alpha\beta} - C_\beta)(1 - z_\alpha) \right. \\ \left. + \sum_{i \in \mathcal{V}_{\alpha\beta}} (C_{\alpha,\beta} - C_\alpha)t_i(1 - z_\beta) + \sum_{i \in \mathcal{V}_{\alpha\beta}} (C_{\alpha\beta} - C_\beta)(1 - t_i)z_\alpha \right]. \quad (12)$$

*Proof.* See appendix. This binary function is pairwise submodular and thus can be solved efficiently using graph cuts.

### 3.2 $\alpha$ -Expansion Moves

In an  $\alpha$ -expansion move every random variable can either retain its current label or transition to label  $\alpha$ . One iteration of the algorithm involves making moves for all  $\alpha$  in

$\mathcal{L}$  successively. The transformation function  $T_\alpha(x_i, t_i)$  for an  $\alpha$ -expansion move transforms the label of a random variable  $x_i$  as:

$$T_\alpha(x_i, t_i) = \begin{cases} x_i & \text{if } t_i = 0 \\ \alpha & \text{if } t_i = 1. \end{cases} \tag{13}$$

To derive a graph-construction that approximates the true cost of an  $\alpha$ -expansion move we rewrite  $C(L)$  as:

$$C(L) = \sum_{B \subseteq L} k_B, \tag{14}$$

where the coefficients  $k_B$  are calculated recursively as:

$$k_B = C(B) - \sum_{B' \subset B} k_{B'}. \tag{15}$$

As a simplifying assumption, let us first assume there is no variable currently taking label  $\alpha$ . Let  $A$  be set of labels currently present in the image and  $\delta_l(\mathbf{t})$  be set to 1 if label  $l$  is present in the image after the move and 0 otherwise. Then:

$$\delta_\alpha(\mathbf{t}) = \begin{cases} 1 & \text{if } \exists i \in \mathcal{V} \text{ s.t. } t_i = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

$$\forall l \in A, \delta_l(\mathbf{t}) = \begin{cases} 1 & \text{if } \exists i \in \mathcal{V}_l \text{ s.t. } t_i = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

The  $\alpha$ -expansion move energy of  $C(L(\mathbf{x}))$  can be written as:

$$E(\mathbf{t}) = E_{new}(\mathbf{t}) - E_{old} = \sum_{B \subseteq A \cup \{\alpha\}} k_B \prod_{l \in B} \delta_l(\mathbf{t}) - C(A).$$

Ignoring the constant term and decomposing the sum into parts with and without terms dependent on  $\alpha$  we have:

$$E(\mathbf{t}) = \sum_{B \subseteq A} k_B \prod_{l \in B} \delta_l(\mathbf{t}) + \sum_{B \subseteq A} k_{B \cup \{\alpha\}} \delta_\alpha(\mathbf{t}) \prod_{l \in B} \delta_l(\mathbf{t}). \tag{18}$$

As either  $\alpha$  or all subsets  $B \subseteq A$  are present after any move, the following statement holds:

$$\delta_\alpha(\mathbf{t}) \prod_{l \in B} \delta_l(\mathbf{t}) = \delta_\alpha(\mathbf{t}) + \prod_{l \in B} \delta_l(\mathbf{t}) - 1. \tag{19}$$

Replacing the term  $\delta_\alpha(\mathbf{t}) \prod_{l \in B} \delta_l(\mathbf{t})$  and disregarding new constant terms, equation (18) becomes:

$$E(\mathbf{t}) = \sum_{B \subseteq A} k_{B \cup \{\alpha\}} \delta_\alpha(\mathbf{t}) + \sum_{B \subseteq A} (k_B + k_{B \cup \{\alpha\}}) \prod_{l \in B} \delta_l(\mathbf{t}) = k'_\alpha \delta_\alpha(\mathbf{t}) + \sum_{B \subseteq A} k'_B \prod_{l \in B} \delta_l(\mathbf{t}), \tag{20}$$

where  $k'_\alpha = \sum_{B \subseteq A} k_{B \cup \{\alpha\}} = C(B \cup \{\alpha\}) - C(B)$  and  $k'_B = k_B + k_{B \cup \{\alpha\}}$ .

$E(\mathbf{t})$  is, in general, a higher-order non-submodular energy, and intractable. However, when proposing moves we can use the procedure described in [21][24] and over-estimate the cost of moving from the current solution. If  $k'_B \geq 0$  the term  $k_B \prod_{l \in B} \delta_l(\mathbf{t})$  is supermodular, and can be over estimated by a linear function  $E_B(\mathbf{t})$  such that  $E_B(A) = k'_B \prod_{l \in B} \delta_l(A)$  and  $E_B(\mathbf{t}) \geq k'_B \prod_{l \in B} \delta_l(\mathbf{t})$  i.e. :

$$E_B(\mathbf{t}) = k'_B \prod_{l \in B} \delta_l(\mathbf{t}) \leq k'_B \sum_{l \in B} \rho_l^B \delta_l(\mathbf{t}), \quad (21)$$

where  $\rho_l^B \geq 0$  and  $\sum_{l \in B} \rho_l^B = 1$ . While  $\rho_l^B$  can always be chosen such that the moves proposed are guaranteed to outperform any particular  $\alpha\beta$  swap [24], in practise we set  $\rho_l^B = 1/|B|$ .

For  $k'_B \leq 0$  we overestimate:

$$k'_B \prod_{l \in B} \delta_l(\mathbf{t}) \leq k'_B - k'_B \sum_{l \in B} (1 - \delta_l(\mathbf{t})) = (1 - |B|)k'_B + k'_B \sum_{l \in B} \rho_l^B \delta_l(\mathbf{t}), \quad (22)$$

where  $\rho_l^B = 1$ . Both of these over-estimations are equal to the original move energy for the initial solution. This guarantees that the energy after the move will not increase and must eventually converge. Ignoring new constant terms the move energy becomes:

$$\begin{aligned} E'(\mathbf{t}) &= k'_\alpha \delta_\alpha + \sum_{B \subseteq A} k'_B \sum_{l \in B} \rho_l^B \delta_l(\mathbf{t}) = k'_\alpha \delta_\alpha + \sum_{l \in A} \delta_l(\mathbf{t}) \sum_{B \subseteq A \setminus \{l\}} k'_{B \cup \{l\}} \rho_l^{B \cup \{l\}} \\ &= k'_\alpha \delta_\alpha + \sum_{l \in A} k''_l \delta_l(\mathbf{t}), \end{aligned} \quad (23)$$

where  $k''_l = \sum_{B \subseteq A \setminus \{l\}} k'_{B \cup \{l\}} \rho_l^{B \cup \{l\}}$ . Coefficients  $k''_l$  are non-negative, as  $\forall B \subseteq A, l \in B : k'_{B \cup \{l\}} \rho_l^{B \cup \{l\}} \geq 0$ , while coefficient  $k'_\alpha$  is non-negative for all  $C(L)$  that are monotonically increasing with respect to  $L$ .

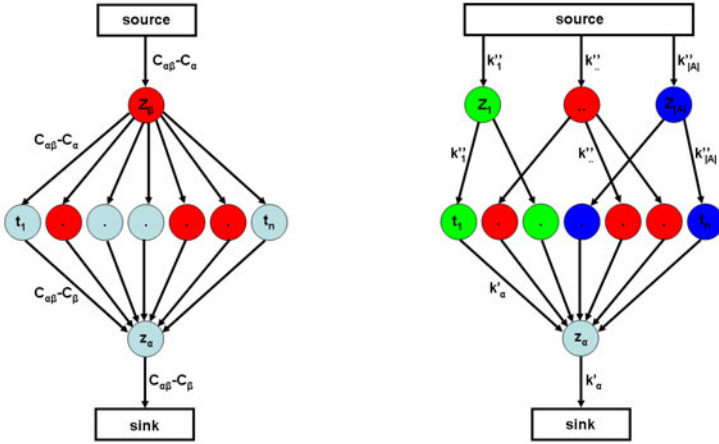
**Lemma 2.** For all  $C(L)$  monotonically increasing with respect to  $L$  the move energy can be represented as a binary pairwise graph with  $|A|$  auxiliary variables  $\mathbf{z}$  as:

$$E'(\mathbf{t}) = \min_{\mathbf{z}} \left[ k'_\alpha (1 - z_\alpha) + \sum_{l \in A} k''_l z_l + \sum_{i \in \mathcal{V}} k'_\alpha (1 - t_i) z_\alpha + \sum_{l \in A} \sum_{i \in \mathcal{V}_l} k''_l t_i (1 - z_l) \right], \quad (24)$$

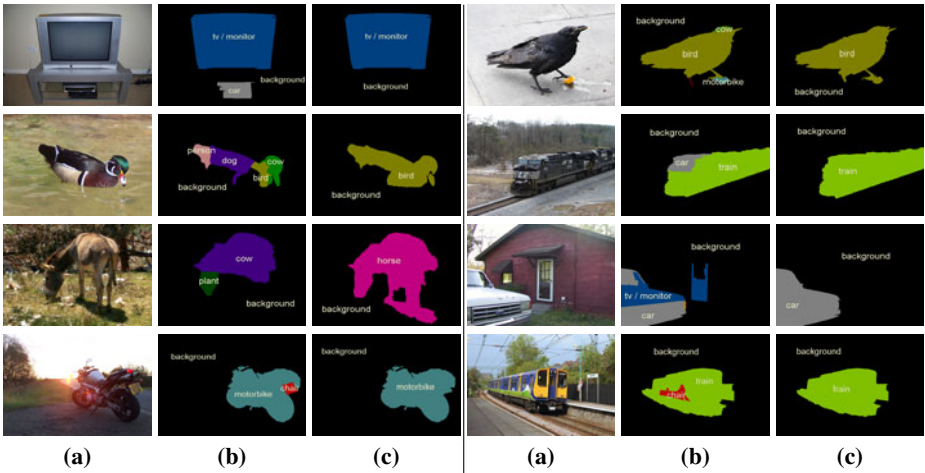
where  $\mathcal{V}_l$  is the set of pixels currently taking label  $l$ .

*Proof.* See appendix. This binary function is pairwise submodular and thus can be solved efficiently using graph cuts.

For co-occurrence potentials monotonically increasing with respect to  $L(\mathbf{x})$  the problem can be modelled using one binary variable  $z_l$  per class indicating the presence of pixels of that class in the labelling, infinite edges for  $x_i = l$  and  $z_l = 0$  and hyper-graph over all  $z_l$  modelling  $C(L(\mathbf{x}))$ . The derived  $\alpha$ -expansion construction can be seen as a graph taking into account costs over all auxiliary variables  $z_l$  for each move and over-estimating the hyper-graph energy using unary potentials. Note that the energy approximation is exact, unless existing classes are removed from the labelling. Consequentially, the only effect our approximation can have on the final labelling is to over estimate the number of classes present in an image. In practice the solutions found by expansion were generally local optima of the exact swap moves.



**Fig. 3.** Graph construction for  $\alpha\beta$ -swap and  $\alpha$ -expansion move. In  $\alpha\beta$ -swap variable  $x_i$  will take the label  $\alpha$  if corresponding  $t_i$  are tied to the sink after the st-mincut and  $\beta$  otherwise. In  $\alpha$ -expansion variable  $x_i$  changes the label to  $\alpha$  if it is tied to the sink after the st-mincut and remains the same otherwise. Colours represent the label of the variables before the move.



**Fig. 4. Best viewed in colour:** (a) Typical images taken from the VOC-2009 data set [29]; (b) A labelling based upon a pixel based random field model [17] that does not take into account co-occurrence; (c) A labelling of the same model using co-occurrence statistics. Note that the co-occurrence potentials perform in a similar way across different data sets, suppressing the smaller classes (see also figure 1) if they appear together in an uncommon combination with other classes such as a car with a monitor, a train with a chair or a dog with a bird. This results in a qualitative rather than quantitative difference.

## 4 Experiments

We performed a controlled test evaluating the performance of CRF models both with and without co-occurrence potentials. As a base line we used the segment-based CRF and the associative hierarchical random field (AHRF) model proposed in [17] and the inference method [26], which currently offers state of the art performance on the MSRC data set [29]. On the VOC data set, the baseline also makes use of the detector potentials of [18]. The costs  $C(L)$  were created from the training set as follows: let  $M$  be the number of images,  $\mathbf{x}^{(m)}$  the ground truth labelling of an image  $m$  and

$$z_l^{(m)} = \delta(l \in L(\mathbf{x}^{(m)})) \quad (25)$$

an indicator function for label  $l$  appearing in an image  $m$ . The associated cost was trained as:

$$C(L) = -w \log \frac{1}{M} \left( 1 + \sum_{m=1}^M \prod_{l \in L} z_l^{(m)} \right), \quad (26)$$

where  $w$  is the weight of the co-occurrence potential. The form guarantees, that  $C(L)$  is monotonically increasing with respect to  $L$ . To avoid over-fitting we approximated the potential  $C(L)$  as a second order function:

$$C'(L) = \sum_{l \in L} c_l + \sum_{k, l \in L, k < l} c_{kl}, \quad (27)$$

where  $c_l$  and  $c_{lk}$  minimise the mean-squared error between  $C(L)$  and  $C'(L)$ .

On the MSRC data set we observed a 3% overall and 4% average per class increase in the recall and 6% in the intersection vs. union measure with the of the segment-based CRF and a 1% overall, 2% average per class and 2% in the intersection vs. union measure with the AHRF. The comparison on the VOC2009 data set was performed on the validation set, as the test set is not published and the number of permitted submissions is limited. Performance improved by 3.5% in the intersection vs. union measure used in the challenge. The performance on the test set was 32.11% which is comparable with current state-of-the-art methods. Results for both data sets are given in tables 5 and 6.

By adding a co-occurrence cost into the CRF we observe constant improvement in pixel classification for almost all classes in all measures. In accordance with desiderata (iv), the co-occurrence potentials tend to suppress uncommon combination of classes and produce more coherent images in the labels space. This results in a qualitative rather than quantitative difference. Although the unary potentials already capture textual context [29], the incorporation of co-occurrence potentials leads to a significant improvement in accuracy.

It is not computationally feasible to perform a direct comparison between the work [22] and our potentials, as the AHRF model is defined over individual pixels, and it is not possible to minimise the resulting fully connected graph which would contain approximately  $4 \times 10^{10}$  edges. Similarly, without their scene classification potentials it was not possible to do a like for like comparison with [31].

Average running time on the MSRC data set without co-occurrence was 5.1s in comparison to 16.1s with co-occurrence cost. On the VOC2009 data set the average times

were 107s and 388s for inference without respectively with co-occurrence costs. We compared the performance of  $\alpha$ -expansion with LP relaxation using solver given in [1] for general co-occurrence potential on the sub-sampled images [16]. Both methods produced similar results in terms of energy, however  $\alpha$ -expansion was approximately 42,000 times faster.

	Global	Average	Building	Grass	Tree	Cow	Sheep	Sky	Aeroplane	Water	Face	Car	Bicycle	Flower	Sign	Bird	Book	Chair	Road	Cat	Dog	Body	Boat
Segment CRF	77	64	70	95	78	55	76	95	63	81	76	67	72	73	82	<b>35</b>	72	17	88	29	62	45	17
Segment CRF with CO	<b>80</b>	<b>68</b>	<b>77</b>	<b>96</b>	<b>80</b>	<b>69</b>	<b>82</b>	<b>98</b>	<b>69</b>	<b>82</b>	<b>79</b>	<b>75</b>	<b>75</b>	<b>81</b>	<b>85</b>	<b>35</b>	<b>76</b>	<b>17</b>	<b>89</b>	<b>25</b>	<b>61</b>	<b>50</b>	<b>22</b>
Hierarchical CRF	86	75	81	<b>96</b>	87	72	84	<b>100</b>	<b>77</b>	<b>92</b>	86	<b>87</b>	87	95	95	27	<b>85</b>	33	<b>93</b>	43	<b>80</b>	62	17
Hierarchical CRF with CO	<b>87</b>	<b>77</b>	<b>82</b>	95	<b>88</b>	<b>73</b>	<b>88</b>	<b>100</b>	<b>83</b>	<b>92</b>	<b>88</b>	<b>87</b>	<b>88</b>	<b>96</b>	<b>96</b>	27	<b>85</b>	<b>37</b>	<b>93</b>	<b>49</b>	<b>80</b>	<b>65</b>	20

**Fig. 5.** Quantitative results on the MSRC data set, average per class recall measure, defined as  $\frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$ . Incorporation of co-occurrence potentials led to a constant improvement for almost every class.

	Average	Background	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dining table	Dog	Horse	Motor bike	Person	Potted plant	Sheep	Sofa	Train	TV/monitor
Hierarchical CRF	27.3	77.7	38.3	9.6	<b>24.0</b>	35.8	<b>31.0</b>	59.2	36.5	21.2	8.3	<b>1.7</b>	22.7	<b>14.3</b>	17.0	26.7	21.1	15.5	<b>16.3</b>	14.6	48.5	33.1
Hierarchical CRF with CO	<b>30.8</b>	<b>82.3</b>	<b>49.3</b>	<b>11.8</b>	19.3	<b>37.7</b>	30.8	<b>63.2</b>	<b>46.0</b>	<b>23.7</b>	<b>10.0</b>	0.5	<b>23.1</b>	14.1	<b>22.4</b>	<b>33.9</b>	<b>35.7</b>	<b>18.4</b>	12.1	<b>22.5</b>	<b>53.1</b>	<b>37.5</b>

**Fig. 6.** Quantitative analysis of VOC2009 results on validation set, intersection vs. union measure, defined as  $\frac{\text{True Positive}}{\text{True Positive} + \text{False Negative} + \text{False Positive}}$ . Incorporation of co-occurrence potential led to labellings, which visually look more coherent, but are not necessarily correct. Quantitatively the performance improved significantly, on average by 3.5% per class.

## 5 Conclusion

The importance of co-occurrence statistics has been well established [31][22][6]. In this work we have examined the use of co-occurrence statistics and how they might be incorporated into a global energy or likelihood model such as a conditional random field. We have discovered that they can naturally be encoded by the use of higher order cliques, without a significant computational overhead. Our new framework provides significant advantages over state of the art approaches including efficient scalable inference. We performed a controlled test evaluating the performance of CRF models both with and without co-occurrence potentials and the incorporation of these potentials results in quantitatively better and visually more coherent labellings.

## References

1. Benson, H.Y., Shanno, D.F.: An exact primal—dual penalty method approach to warmstarting interior-point methods for linear programming. *Comput. Optim. Appl.* (2007)
2. Borenstein, E., Malik, J.: Shape guided object segmentation. In: *CVPR* (2006)

3. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI (2001)
4. Choi, M.J., Lim, J.J., Torralba, A., Willsky, A.S.: Exploiting hierarchical context on a large database of object categories. In: CVPR (2010)
5. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. PAMI (2002)
6. Csurka, G., Perronnin, F.: A simple high performance approach to semantic segmentation. In: BMVC 2008 (2008)
7. Delong, A., Osokin, A., Isack, H., Boykov, Y.: Fast approximate energy minimization with label costs. In: CVPR (2010)
8. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV (2004)
9. Galleguillos, C., Rabinovich, A., Belongie, S.: Object categorization using co-occurrence, location and appearance. In: CVPR (2008)
10. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV (2009)
11. Heitz, D.K.G.: Learning spatial context: Using stuff to find things. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 30–43. Springer, Heidelberg (2008)
12. Hoiem, D., Rother, C., Winn, J.M.: 3d layoutcrf for multi-view object class recognition and segmentation. In: CVPR (2007)
13. Kohli, P., Ladicky, L., Torr, P.H.: Robust higher order potentials for enforcing label consistency. In: CVPR (2008)
14. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. PAMI (2006)
15. Kolmogorov, V., Rother, C.: Comparison of energy minimization algorithms for highly connected graphs. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 1–15. Springer, Heidelberg (2006)
16. Ladicky, L., Russell, C., Kohli, P., Torr, P.H.: Graph Cut based Inference with Co-occurrence Statistics — Technical report (2010)
17. Ladicky, L., Russell, C., Kohli, P., Torr, P.H.: Associative hierarchical crfs for object class image segmentation. In: ICCV (2009)
18. Ladicky, L., Russell, C., Sturges, P., Alahari, K., Torr, P.H.: What, where and how many? Combining object detectors and CRFs. In: ECCV (2010)
19. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labelling sequence data. In: ICML (2001)
20. Larlus, D., Jurie, F.: Combining appearance models and markov random fields for category level object segmentation. In: CVPR (2008)
21. Narasimhan, M., Bilmes, J.A.: A submodular-supermodular procedure with applications to discriminative structure learning. In: UAI (2005)
22. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in context. In: ICCV (2007)
23. Ren, X., Fowlkes, C., Malik, J.: Mid-level cues improve boundary detection. Technical Report UCB/CSD-05-1382, Berkeley (March 2005)
24. Rother, C., Kumar, S., Kolmogorov, V., Blake, A.: Digital tapestry. In: CVPR (2005)
25. Russell, B., Freeman, W., Efros, A., Sivic, J., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: CVPR (2006)
26. Russell, C., Ladicky, L., Kohli, P., Torr, P.H.: Exact and approximate inference in associative hierarchical networks using graph cuts. In: UAI (2010)



27. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. In: Adaptive Computation and Machine Learning. MIT Press, Cambridge (2001)
28. Shi, J., Malik, J.: Normalized cuts and image segmentation. PAMI (2000)
29. 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, Part I. LNCS, vol. 3951, pp. 1–15. Springer, Heidelberg (2006)
30. 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)
31. Torralba, A., Murphy, K.P., Freeman, W.T., Rubin, M.A.: Context-based vision system for place and object recognition. In: Proceedings of Computer Vision (2003)
32. Toyoda, T., Hasegawa, O.: Random field model for integration of local information and global information. PAMI (2008)
33. Weiss, Y., Freeman, W.: On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. Transactions on Information Theory (2001)
34. Yang, L., Meer, P., Foran, D.J.: Multiple class segmentation using a unified framework over mean-shift patches. In: CVPR (2007)

## Appendix

*Lemma 1 Proof.* First we show that:

$$\begin{aligned}
 E_\alpha(\mathbf{t}) &= \min_{z_\alpha} \left[ (C_{\alpha\beta} - C_\beta)(1 - z_\alpha) + \sum_{i \in \mathcal{V}_{\alpha\beta}} (C_{\alpha\beta} - C_\beta)(1 - t_i)z_\alpha \right] \\
 &= \begin{cases} 0 & \text{if } \forall i \in \mathcal{V}_{\alpha\beta} : t_i = 1, \\ C_{\alpha\beta} - C_\beta & \text{otherwise.} \end{cases} \quad (28)
 \end{aligned}$$

If  $\forall i \in \mathcal{V}_{\alpha\beta} : t_i = 1$  then  $\sum_{i \in \mathcal{V}_{\alpha\beta}} (C_{\alpha\beta} - C_\beta)(1 - t_i)z_\alpha = 0$  and the minimum cost cost 0 occurs when  $z_\alpha = 1$ . If  $\exists i \in \mathcal{V}_{\alpha\beta}, t_i = 0$  the minimum cost labelling occurs when  $z_\alpha = 0$  and the minimum cost is  $C_{\alpha\beta} - C_\beta$ .

Similarly:

$$\begin{aligned}
 E_\beta(\mathbf{t}) &= \min_{z_\beta} \left[ (C_{\alpha\beta} - C_\alpha)z_\beta + \sum_{i \in \mathcal{V}_{\alpha\beta}} (C_{\alpha,\beta} - C_\alpha)t_i(1 - z_\beta) \right] \\
 &= \begin{cases} 0 & \text{if } \forall i \in \mathcal{V}_{\alpha\beta} : t_i = 0, \\ C_{\alpha\beta} - C_\alpha & \text{otherwise.} \end{cases} \quad (29)
 \end{aligned}$$

By inspection, if  $\forall i \in \mathcal{V}_{\alpha\beta} : t_i = 0$  then  $\sum_{i \in \mathcal{V}_{\alpha\beta}} (C_{\alpha,\beta} - C_\alpha)t_i(1 - z_\beta) = 0$  and the minimum cost cost 0 occurs when  $z_\beta = 0$ . If  $\exists i \in \mathcal{V}_{\alpha\beta}, t_i = 1$  the minimum cost labelling occurs when  $z_\beta = 1$  and the minimum cost is  $C_{\alpha\beta} - C_\alpha$ .

For all three cases (all pixels take label  $\alpha$ , all pixels take label  $\beta$  and mixed labelling)  $E(\mathbf{t}) = E_\alpha(\mathbf{t}) + E_\beta(\mathbf{t}) + C_\alpha + C_\beta - C_{\alpha\beta}$ . The construction of the  $\alpha\beta$ -swap move is similar to the Robust  $P^N$  model [13].  $\square$

See figure 3 for graph construction.

*Lemma 2 Proof.* Similarly to the  $\alpha\beta$ -swap proof we can show:

$$E_\alpha(\mathbf{t}) = \min_{z_\alpha} \left[ k'_\alpha(1 - z_\alpha) + \sum_{i \in \mathcal{V}} k'_\alpha(1 - t_i)z_\alpha \right] = \begin{cases} k'_\alpha & \text{if } \exists i \in \mathcal{V} \text{ s.t. } t_i = 0, \\ 0 & \text{otherwise.} \end{cases} \quad (30)$$

If  $\exists i \in \mathcal{V} \text{ s.t. } t_i = 0$ , then  $\sum_{i \in \mathcal{V}} k'_\alpha(1 - t_i) \geq k'_\alpha$ , the minimum is reached when  $z_\alpha = 0$  and the cost is  $k'_\alpha$ .

If  $\forall i \in \mathcal{V} : t_i = 1$  then  $k'_\alpha(1 - t_i)z_\alpha = 0$ , the minimum is reached when  $z_\alpha = 1$  and the cost becomes 0.

For all other  $l \in A$ :

$$E_b(\mathbf{t}) = \min_{z_l} \left[ k''_l z_l + \sum_{i \in \mathcal{V}_l} k''_l t_i(1 - z_l) \right] = \begin{cases} k''_l & \text{if } \exists i \in \mathcal{V}_l \text{ s.t. } t_i = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (31)$$

If  $\exists i \in \mathcal{V}_l \text{ s.t. } t_i = 1$ , then  $\sum_{i \in \mathcal{V}_l} k''_l t_i \geq k''_l$ , the minimum is reached when  $z_l = 1$  and the cost is  $k''_l$ .

If  $\forall i \in \mathcal{V}_l : t_i = 0$  then  $\sum_{i \in \mathcal{V}_l} k''_l t_i(1 - z_l) = 0$ , the minimum is reached when  $z_l = 1$  and the cost becomes 0.

By summing up the cost  $E_\alpha(\mathbf{t})$  and  $|A|$  costs  $E_l(\mathbf{t})$  we get  $E'(\mathbf{t}) = E_\alpha(\mathbf{t}) + \sum_{l \in A} E_l(\mathbf{t})$ . If  $\alpha$  is already present in the image  $k'_\alpha = 0$  and edges with this weight and variable  $z_\alpha$  can be ignored.  $\square$

See figure 3 for graph construction.

# Ambrosio-Tortorelli Segmentation of Stochastic Images

Torben Pätz<sup>1,2</sup> and Tobias Preusser<sup>1,2</sup>

<sup>1</sup> School of Engineering and Science, Jacobs University Bremen

<sup>2</sup> Fraunhofer MEVIS, Bremen, Germany

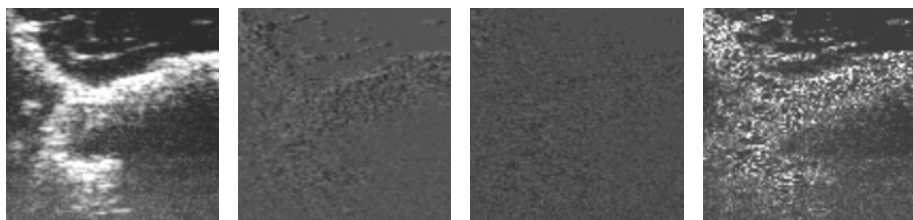
**Abstract.** We present an extension of the classical Ambrosio-Tortorelli approximation of the Mumford-Shah approach for the segmentation of images with uncertain gray values resulting from measurement errors and noise. Our approach yields a reliable precision estimate for the segmentation result, and it allows to quantify the robustness of edges in noisy images and under gray value uncertainty. We develop an ansatz space for such images by identifying gray values with random variables. The use of these stochastic images in the minimization of energies of Ambrosio-Tortorelli type leads to stochastic partial differential equations for the stochastic smoothed image and a stochastic phase field for the edge set. For their discretization we utilize the generalized polynomial chaos expansion and the generalized spectral decomposition (GSD) method. We demonstrate the performance of the method on artificial data as well as real medical ultrasound data.

**Keywords:** Image processing, segmentation, uncertainty, stochastic images, stochastic partial differential equation, polynomial chaos, generalized spectral decomposition.

## 1 Introduction

In many applications images are used for quantitative measurements, e.g. to determine the size or distance of objects. As image acquisition itself (e.g. by digital camera, CT, MR or Ultrasound) involves measurements of physical or chemical quantities or properties it is good scientific practice that these measurements are equipped with error estimates and that these error estimates are propagated through all analysis steps, including quantitative image processing. The goal is a reliable precision estimate for the final result. In quantitative medical imaging this for example can support the evaluation of the treatment response in chemotherapy. There the growth or shrinkage of tumors must be detected robustly on base of noisy contrast enhanced CT scans. As a matter of fact small measurement errors due to noise and uncertainty in the gray values can result in huge variations in the computed tumor volume, thus being a source for erroneous therapy-response indications.

Quantitative image processing is often related to the segmentation of an object inside an image. The main idea is to detect the shape of an object inside an image



**Fig. 1.** From left to right: The first mode (=mean), second mode, fifth mode and the variance of a stochastic image are shown

and to separate it from the background. In the past, a multitude of methods based on PDEs have been developed. Among them are level set (sharp interface) and phase field (diffuse interface) approaches, which use implicit representations of the object boundaries that are computationally much easier to handle than explicit representations.

In [10] a speed function for the level set evolution is proposed, which depends on the image gradient. The evolution stops when the level set reaches an edge inside the image. This method was improved by Caselles et. al. [4] by introducing an additional term, which forces the level set to stay at the boundary. The idea of the Chan and Vese approach [5] is to segment homogeneous regions inside an image. An evolution equation is solved until the level set separates homogeneous regions of the image. This allows to segment objects with and without sharp edges. The well known Mumford-Shah approach [12] is based on the minimization of an energy functional, which measures the smoothness of the segmented objects as well as the length of the object boundaries. An often used regularization of the Mumford-Shah functional is the method proposed by Ambrosio-Tortorelli [1] leading to a phase field model for the description of object boundaries.

Error propagation is very difficult for classical image processing algorithms and in particular for the level set or phase field segmentation methods mentioned above. In the literature a lot of authors deal with error estimates, which have several restrictions: Weber et. al. [17] presented a method where the input data is presumed to be Gaussian distributed. Nestares et. al. [13] were able to derive bounds for the error and Bruhn et. al. [2] derived confidence measures for the error. In [15] a method is presented, which assumes values of the image pixels not to have fixed gray values but distributions of gray values. Thus, pixels are random variables (RVs), which model the errors in the image acquisition process. An image containing such RVs as pixels/voxels is then called a stochastic image. A few modes of a stochastic image are pictured in Fig. 1.

In the work presented here we extend this approach of stochastic images and combine it with Mumford-Shah segmentation in the spirit of the Ambrosio-Tortorelli phase field approximation. For an input image with uncertain gray values our approach provides a stochastic edge representation in the form of a stochastic phase field as well as a stochastic image as the representation of the smoothed input image. It allows for precise error estimates beyond the assumption of Gaussian gray value distributions. In fact, the evaluation of the stochastic

modes of the phase field allows e.g. to estimate the variance of the edge location or the confidence for the presence of edges at certain locations for arbitrary noise models and distributions of gray values.

The use of the notion of stochastic images in variational image processing leads to stochastic partial differential equations (SPDE). The numerical solution of SPDEs is a challenging problem, because intrusive methods like the stochastic finite element method (SFEM) [8] lead to high dimensional systems of equations, which are difficult to treat on contemporary hardware. We utilize the recently developed generalized spectral decomposition (GSD) [14], which allows to break down the systems of equations into a series of smaller systems by choosing optimal small subspaces in the stochastic dimension. This results in an enormous speedup of the computation, a saving of memory, and in an algorithm, which is much faster than classical sampling techniques like Monte Carlo.

## 2 Stochastic Images

It is popular in PDE based image processing to model an image  $f : D \rightarrow \mathbb{R}$  on a domain  $D \subset \mathbb{R}^d$ ,  $d = 2, 3$  using a finite element space and a representation

$$f(x) = \sum_{i \in \mathcal{I}} f^i P_i(x) , \quad (1)$$

where  $f^i \in \mathbb{R}$  is the value of the  $i$ -th pixel from the pixel set  $\mathcal{I}$  and  $P_i$  the shape function (e.g. tent-function) of the  $i$ -th pixel. In a stochastic image a single pixel has no longer a fixed value. Instead it depends on a vector of RVs  $\boldsymbol{\xi}(\omega) = (\xi_1(\omega), \dots, \xi_n(\omega))$  and on a random event  $\omega \in \Omega$ . Here  $\Omega$  denotes an event space,  $\mathcal{A} \subset 2^\Omega$  a  $\sigma$ -algebra and  $\Pi$  a probability measure. Note that the concept of stochastic images can also be combined with other spacial discretizations, e.g. finite difference schemes.

### 2.1 Polynomial Chaos Expansion

Based on the fundamental work of Wiener [18], Xiu and Karniadakis [20] developed the generalized polynomial chaos (gPC) expansion for the representation of a RV with finite second-order moments by a polynomial basis.

Following Cameron/Martin [3], every RV  $X(\omega) \in L^2(\Omega, \mathcal{A}, \Pi)$  can be represented by

$$X(\omega) = \sum_{\alpha=1}^{\infty} a_{\alpha} \Psi^{\alpha}(\hat{\xi}(\omega)) , \quad (2)$$

where  $\hat{\xi} = (\xi_1, \xi_2, \dots)$  is a sequence of RVs with known probability density function  $\rho_j$  and  $\Psi^{\alpha}$  are polynomials in  $\hat{\xi}$  forming a basis of  $L^2(\Omega, \mathcal{A}, \Pi)$ . For the numerical treatment an approximation with prescribed polynomial degree  $p$  and a fixed number of RVs  $\boldsymbol{\xi} = (\xi_1, \dots, \xi_n)$  is chosen, thus

$$X(\omega) \approx \sum_{\alpha=1}^N a_{\alpha} \Psi^{\alpha}(\boldsymbol{\xi}(\omega)) . \quad (3)$$

This representation involves a mapping  $\xi : \Omega \rightarrow \Gamma$  of events  $\omega \in \Omega$  to  $\xi(\omega) \in \Gamma$ , where  $\Gamma = \times_{j=1}^n \xi_j(\Omega)$  is finite dimensional.

The number  $N$  of basis functions depends on the number  $n$  of RVs and the maximal polynomial degree  $p$  of the approximation. As usual for a polynomial basis the number of basis functions is given by  $N = \binom{n+p}{p}$ . Thus, the number of basis functions grows rapidly with the number of RVs  $n$  and the polynomial degree  $p$  of the approximation.

It is most convenient to choose polynomials  $\Psi^\alpha$  which are pairwise orthogonal with respect to the corresponding probability measure of the  $\xi_j$ . Thus, in the case of Gaussian RVs  $\xi_j$  the  $\Psi^\alpha$  are products of one-dimensional Hermite polynomials. In the case of uniformly distributed RVs the  $\Psi^\alpha$  are products of Legendre polynomials. In our work presented here we use uniformly distributed RVs  $\xi_j$  involving Legendre polynomials. For arithmetic operations needed for the use of the gPC expansion in numerical schemes we use the methods from [6].

The expectation, variance and analogously higher stochastic moments of the approximated RV  $X(\omega)$  are evaluated as

$$\mathbb{E}(X) \approx \int_{\Gamma} \sum_{\alpha=1}^N a_\alpha \Psi^\alpha(\xi) d\Pi_\xi, \quad \text{Var}(X) \approx \int_{\Gamma} \left( \sum_{\alpha=1}^N a_\alpha \Psi^\alpha(\xi) - \mathbb{E}(X) \right)^2 d\Pi_\xi, \quad (4)$$

where  $d\Pi_\xi = \prod_{j=1}^n \rho_j(\xi_j) d\xi_j$  is the transformed probability measure.

### 2.2 Polynomial Chaos for Stochastic Images

Following [15], the representation of an image whose pixels values are RVs is obtained from (1) by replacing the fixed  $f^i$  by RVs  $f^i(\xi)$ , thus

$$f(x, \xi) = \sum_{i \in \mathcal{I}} f^i(\xi) P_i(x). \quad (5)$$

Note that here and in the following we omit denoting the dependence of  $\xi$  on  $\omega$  for reasons of simplicity of the presentation. The gPC expansion (3) allows to approximate any second order RV  $f^i(\xi)$  by a weighted sum of orthogonal multidimensional polynomials. This leads to

$$f(x, \xi) = \sum_{i \in \mathcal{I}} \sum_{\alpha=1}^N f_\alpha^i \Psi^\alpha(\xi) P_i(x) \quad (6)$$

as the representation of a stochastic image, i.e. an image whose gray values are RVs. For fixed  $\alpha$  we call the coefficient  $f_\alpha^i$  a *stochastic mode* of the pixel  $i$ . The set  $\{f_\alpha^i\}_{i \in \mathcal{I}}$  collects the stochastic modes of all pixels for fixed  $\alpha$ . This set can be visualized as a classical image, which is done in Fig. 1 where three modes of a sample image are shown.

From the gPC expansion (6) it is straight forward to compute stochastic moments of the images. With the use of our orthogonal set of basis functions, the Legendre Polynomials, we have  $\mathbb{E}(\Psi^1) = 1$  and  $\mathbb{E}(\Psi^\alpha \Psi^\beta) = 0$  if  $\alpha \neq \beta$ . Thus, the mean and the variance of a stochastic image are computed as

$$\mathbb{E}(f(x_i, \cdot)) = f_1^i, \quad \text{Var}(f(x_i, \cdot)) = \sum_{\alpha=2}^N (f_\alpha^i)^2 \mathbb{E}((\Psi^\alpha)^2). \quad (7)$$

Other stochastic moments are obtained in a similar way. In Fig. 1 the mean and the variance of a stochastic image are shown.

Note that the representation of stochastic images presented here differs from the one discussed in [15]. There, an image space is used in which every pixel depends on one RV only. However, for many image acquisition processes and image processing methods the assumption that the noise is independent for every pixel is not true, thus we let every pixel depend on a vector of RVs  $\xi$ .

### 2.3 From Samples to Input Distributions

To use the notion of stochastic images developed in the previous sections for image processing, we need to obtain the coefficients of the representation (6) for our image undergoing the analysis. Let  $u^{(1)}, \dots, u^{(M)}$ , with  $u^{(k)} \in \mathbb{R}^r$ ,  $r = |\mathcal{I}|$ , denote sample images, e.g. resulting from repeated acquisition. The goal is to identify these image samples as the samples of some vector of independent RVs  $\mathbf{X}$ . To this end the empirical Karhunen-Loeve decomposition [9] yields

$$u^{(k)} = \bar{u} + \sum_{j=1}^r \sqrt{s_j} U_j X_j^{(k)} \quad , \quad (8)$$

where  $\bar{u}$  is the mean of the input samples. The pairs  $(s_j, U_j)$  for  $j = 1, \dots, r$  are the eigenpairs sorted in descending order of the  $r \times r$  covariance matrix

$$C := \frac{1}{M-1} \sum_{k=1}^M (u^{(k)} - \bar{u})^T (u^{(k)} - \bar{u}) \quad . \quad (9)$$

Moreover, the

$$X_j^{(k)} = (s_j)^{-1/2} U_j^T (u^{(k)} - \bar{u}) \quad (10)$$

are samples of the desired vector of RVs  $\mathbf{X} = (X_1, \dots, X_n)$ , where  $n < r$ .

The estimation of the coefficients of the gPC expansion (3) of the random vector  $\mathbf{X}$  from these samples can be achieved by inverting the discrete empirical cumulative distribution function (CDF)  $F_{X_j}$ , which is based on the samples  $X_j^{(k)}$ . This leads to a staircase-like approximation of the RV  $X_j$ . Following [16] we get  $X_{j,\alpha}$  from the projection on  $\Psi^\alpha$  via

$$X_{j,\alpha} = \mathbb{E}(X_j \Psi^\alpha) = \int_{\Gamma} F_{X_j}^{-1}(F_\xi(y)) \Psi^\alpha(\xi(y)) d\Pi_\xi(y) \quad . \quad (11)$$

Note that the assumption of independence allows us to work with those basis functions, which depend on one RV only, i.e.  $\Psi^\alpha(\xi) = \Psi^\alpha(\xi)$ . The empirical CDF and the empirical inverse of the CDF are obtained by

$$F_{X_j}(x) = \frac{1}{M} \sum_{k=1}^M I(X_j^{(k)} \leq x) \quad , \quad (12)$$

$$F_{X_j}^{-1}(y) = \min \left\{ x \in \{X_j^{(k)}\}_{k=1}^M \mid F_{X_j}(x) \geq y \right\} \quad ,$$

where  $I$  is the indicator function attaining value 1 for true arguments and 0 else. Note that the RVs  $X_j$  are related to the eigenpairs  $(s_j, U_j)$  of the Karhunen-Loeve decomposition via (10). Using the expression for the inverse  $F_{X_j}^{-1}$  together with a numerical quadrature associated with the measure  $\Pi_\xi$  allows to compute the gPC expansion coefficients  $X_j^\alpha$  independently from each other.

We emphasize that the assumption of independence of the RVs  $X_j$  is very strong and in general not true. However, following [16] in particular for the case of few input samples this assumption is reasonable.

Also note that the estimation of the typically dense covariance matrix is only feasible for images of low dimension  $r$ . For large image dimensions we must model the gray value distribution using characteristics of the acquisition process and not via the analysis of samples. This noise modeling is part of ongoing work.

### 3 A Phase Field Model for Segmentation on Stochastic Images

We now focus on the combination of the notion of stochastic images with the segmentation approach in the spirit of Ambrosio and Tortorelli [1].

#### 3.1 Classical Mumford-Shah and Ambrosio-Tortorelli Segmentation

For a given initial image  $u_0$  on the domain  $D$  Mumford and Shah [12] proposed to obtain an edge set  $K \subset D$  and a smooth representation  $u$  of  $u_0$  as the minimizers of the energy

$$E_{\text{MS}}(u, K) := \int_{D \setminus K} (u - u_0)^2 dx + \mu \int_{D \setminus K} |\nabla u|^2 dx + \nu \mathcal{H}^{d-1}(K) \quad , \quad (13)$$

where  $\mu$  and  $\nu$  are positive weights, and  $\mathcal{H}^{d-1}(K)$  the  $d-1$ -dimensional Hausdorff measure. Roughly speaking, the minimizer  $u$  must be an image, which is close to the initial  $u_0$  away from the edges (then  $\int_{D \setminus K} (u - u_0)^2 dx$  is small) and smooth away from the edges (then  $\int_{D \setminus K} |\nabla u|^2 dx$  is small). Moreover the length of edges  $K$  must be small (then  $\mathcal{H}^{d-1}(K)$ , measuring the length of the edge set, is small).

Ambrosio and Tortorelli [1] proposed to approximate the edge set by a phase field  $\phi : D \rightarrow \mathbb{R}$ , i.e. a smooth function that is zero on edges and one away from the edges. To this end they define the energy

$$\begin{aligned} E_{\text{AT}}(u, \phi) &:= E_{\text{fid}}(u) + E_{\text{reg}}(u, \phi) + E_{\text{phase}}(\phi) \\ &:= \int_D (u(x) - u_0(x))^2 dx + \int_D \mu (\phi(x)^2 + k_\varepsilon) |\nabla u(x)|^2 dx \\ &\quad + \int_D \nu \varepsilon |\nabla \phi(x)|^2 + \frac{\nu}{4\varepsilon} (1 - \phi(x))^2 dx \quad . \end{aligned} \quad (14)$$

The first integral ensures closeness of the smoothed image to the original  $u_0$ . The second integral measures smoothness of  $u$  apart from areas where  $\phi$  is small,



and enforces  $\phi$  to be small in the vicinity of edges. The parameter  $k_\varepsilon$  ensures coerciveness of the differential operator and thus existence of solutions, because  $\phi^2$  may vanish. The third integral drives the phase field towards one and ensures small edge sets via the term  $|\nabla\phi|^2$ . The parameter  $\varepsilon$  allows to control the scale of the detected edges.

For the numerical determination of a minimizing pair  $(u, \phi)$  the Euler-Lagrange equations of (14) are solved. Thus we seek  $u, \phi \in H^1(D)$  as the weak solutions of

$$-\operatorname{div}(\mu(\phi^2 + k_\varepsilon)\nabla u) + u = u_0, \quad -\varepsilon\Delta\phi + \left(\frac{1}{4\varepsilon} + \frac{\mu}{2\nu}|\nabla u|^2\right)\phi = \frac{1}{4\varepsilon}. \quad (15)$$

In an implementation both equations can be solved alternately letting either  $u$  or  $\phi$  vary until a fixed point as the joint solution of both equations is reached.

### 3.2 Ambrosio-Tortorelli Segmentation on Stochastic Images

For the segmentation of stochastic images by the phase field approach of Ambrosio and Tortorelli we replace the deterministic  $u$  and  $\phi$  by their corresponding stochastic analogs. The stochastic energy components are then defined as the expectations of the classical components, i.e.

$$\begin{aligned} E_{\text{fid}}^s(u) &:= \mathbb{E}(E_{\text{fid}}) = \int_\Gamma \int_D (u(x, \boldsymbol{\xi}) - u_0(x, \boldsymbol{\xi}))^2 \, dx \, d\Pi_\boldsymbol{\xi} \\ E_{\text{reg}}^s(u, \phi) &:= \mathbb{E}(E_{\text{reg}}) = \int_\Gamma \int_D \mu \left( \phi(x, \boldsymbol{\xi})^2 + k_\varepsilon \right) |\nabla u(x, \boldsymbol{\xi})|^2 \, dx \, d\Pi_\boldsymbol{\xi} \\ E_{\text{phase}}^s(\phi) &:= \mathbb{E}(E_{\text{phase}}) = \int_\Gamma \int_D \nu \varepsilon |\nabla \phi(x, \boldsymbol{\xi})|^2 + \frac{\nu}{4\varepsilon} (1 - \phi(x, \boldsymbol{\xi}))^2 \, dx \, d\Pi_\boldsymbol{\xi} \end{aligned} \quad (16)$$

and we define the stochastic energy as the sum of these, i.e.

$$E_{AT}^s(u, \phi) = E_{\text{fid}}^s(u) + E_{\text{reg}}^s(u, \phi) + E_{\text{phase}}^s(\phi). \quad (17)$$

The Euler-Lagrange equations of the energy are obtained from the first variation of the above integrals. Since the stochastic energies (16) are just the expectations of the classical energies (14) the computations are straight forward and completely analog to the deterministic case. For example, we get for a test function  $\theta : D \times \Gamma \rightarrow \mathbb{R}$

$$\begin{aligned} \frac{d}{dt} E_{\text{fid}}^s(u+t\theta) \Big|_{t=0} &= \frac{d}{dt} \int_\Gamma \int_D \left( u(x, \boldsymbol{\xi}) + t\theta(x, \boldsymbol{\xi}) - u_0(x, \boldsymbol{\xi}) \right)^2 \, dx \, d\Pi_\boldsymbol{\xi} \Big|_{t=0} \\ &= \int_\Gamma \int_D 2 \left( u(x, \boldsymbol{\xi}) - u_0(x, \boldsymbol{\xi}) \right) \theta(x, \boldsymbol{\xi}) \, dx \, d\Pi_\boldsymbol{\xi}. \end{aligned} \quad (18)$$

With analog computations for the remaining energy contributions we arrive at the following system of stochastic partial differential equations: We seek for  $u, \phi : D \times \Gamma \rightarrow \mathbb{R}$  as the weak solutions of

$$\begin{aligned}
 &-\operatorname{div}(\mu(\phi(x, \boldsymbol{\xi})^2 + k_\varepsilon)\nabla u(x, \boldsymbol{\xi})) + u(x, \boldsymbol{\xi}) = u_0(x, \boldsymbol{\xi}) \\
 &-\varepsilon\Delta\phi(x, \boldsymbol{\xi}) + \left(\frac{1}{4\varepsilon} + \frac{\mu}{2\nu}|\nabla u(x, \boldsymbol{\xi})|^2\right)\phi(x, \boldsymbol{\xi}) = \frac{1}{4\varepsilon} .
 \end{aligned} \tag{19}$$

This system is analog to the classical system (15) in which images have been replaced by stochastic images.

### 3.3 Weak Formulation and Discretization

The system (19) contains two elliptic SPDEs, which are supposed to be interpreted in the weak sense. To this end we multiply the equations by a test function  $\theta : D \times \Gamma \rightarrow \mathbb{R}$ , integrate over  $\Gamma$  with respect to the corresponding probability measure and integrate by parts over the physical domain  $D$ . For the first equation in (19) this leads us to

$$\begin{aligned}
 \int_{\Gamma} \int_D \mu(\phi(x, \boldsymbol{\xi})^2 + k_\varepsilon) \nabla u(x, \boldsymbol{\xi}) \cdot \nabla \theta(x, \boldsymbol{\xi}) + u(x, \boldsymbol{\xi})\theta(x, \boldsymbol{\xi}) \, dx \, d\Pi_{\boldsymbol{\xi}} \\
 = \int_{\Gamma} \int_D u_0(x, \boldsymbol{\xi})\theta(x, \boldsymbol{\xi}) \, dx \, d\Pi_{\boldsymbol{\xi}}
 \end{aligned} \tag{20}$$

and to an analog expression for the second part of (19). Here we assume Neumann (natural) boundary conditions for  $u$  and  $\phi$  such that no boundary terms appear in the weak form. For the existence of solutions for these SPDEs, the constant  $k_\varepsilon$  is supposed to ensure the positivity of the diffusion coefficient  $\mu(\phi^2 + k_\varepsilon)$ . In fact, there must exist  $c_{\min}, c_{\max} \in (0, \infty)$  such that

$$P\left(\omega \in \Omega \mid \mu\left(\phi(x, \boldsymbol{\xi}(\omega))^2 + k_\varepsilon\right) \in [c_{\min}, c_{\max}] \forall x \in \overline{D}\right) = 1 . \tag{21}$$

Finally, solutions  $u$  and  $\phi$  will be random fields, i.e. RVs, which are indexed by a spatial coordinate and such that  $u(\cdot, \boldsymbol{\xi}), \phi(\cdot, \boldsymbol{\xi}) \in H^1(D)$  almost sure. Thus, for almost every realization (in the sense of the measure  $\Pi_{\boldsymbol{\xi}}$ ) the stochastic images  $u$  and  $\phi$  have weak derivatives in  $L^2(D)$ .

The weak system (20) is discretized with a substitution of the gPC expansion (6) of the image and the phase field. As test functions products  $P_j(x)\Psi^\beta(\boldsymbol{\xi})$  of spatial basis functions and stochastic basis functions are used. Denoting the vectors of coefficients by  $U^\alpha = (u_\alpha^i)_{i \in \mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|}$  and similarly for the phase field  $\phi$  and the initial image  $u_0$  we get the fully discrete systems

$$\begin{aligned}
 \sum_{\alpha=1}^N (M^{\alpha,\beta} + L^{\alpha,\beta}) U^\alpha &= \sum_{\alpha=1}^N M^{\alpha,\beta} (U_0)^\alpha \quad \forall \beta \in \{1, \dots, N\} \\
 \sum_{\alpha=1}^N (\varepsilon S^{\alpha,\beta} + T^{\alpha,\beta}) \Phi^\alpha &= \sum_{\alpha=1}^N A^\alpha \quad \forall \beta \in \{1, \dots, N\}
 \end{aligned} \tag{22}$$

where  $M^{\alpha,\beta}, L^{\alpha,\beta}, S^{\alpha,\beta}$  and  $T^{\alpha,\beta}$  are the blocks of the system matrix, defined as

$$(M^{\alpha,\beta})_{i,j} = \mathbb{E}(\Psi^\alpha \Psi^\beta) \int_D P_i P_j \, dx, \quad (S^{\alpha,\beta})_{i,j} = \mathbb{E}(\Psi^\alpha \Psi^\beta) \int_D \nabla P_i \cdot \nabla P_j \, dx \tag{23}$$

and

$$\begin{aligned}
 (L^{\alpha,\beta})_{i,j} &= \sum_k \sum_\gamma \mathbb{E}(\Psi^\alpha \Psi^\beta \Psi^\gamma) (\widetilde{\phi^2})_\gamma^k \int_D \nabla P_i \cdot \nabla P_j P_k \, dx, \\
 (T^{\alpha,\beta})_{i,j} &= \sum_k \sum_\gamma \mathbb{E}(\Psi^\alpha \Psi^\beta \Psi^\gamma) u_\gamma^k \int_D P_i P_j P_k \, dx.
 \end{aligned}
 \tag{24}$$

Here,  $(\widetilde{\phi^2})_\gamma^k$  denotes the coefficients of the gPC expansion of the Galerkin projection of  $\phi^2$  onto the image space (cf. [6]). Finally, the right hand side vector is defined as

$$(A^\alpha)_i = \int_\Gamma \Psi^\alpha d\xi \int_D \frac{1}{4\varepsilon} P_i \, dx = \begin{cases} \int_D \frac{1}{4\varepsilon} P_i \, dx & \text{if } \alpha = 1, \\ 0 & \text{else} \end{cases} .
 \tag{25}$$

Note that the expectations of the products of stochastic basis functions involved above can be precomputed in advance, since these do only depend on the choice of basis functions. Analog to the classical finite element method the systems of linear equations can be treated by an iterative solver like the method of conjugate gradients.

### 3.4 Generalized Spectral Decomposition

A significant speedup of the solution process and an enormous reduction of the memory requirements are achieved by selecting suitable sub-spaces and a special basis which captures the dominant stochastic effects. In the GSD [14] the solution  $u$  (analogously  $\phi$ ) is approximated by

$$u(x, \boldsymbol{\xi}) \approx \sum_{j=1}^K \lambda_j(\boldsymbol{\xi}) V_j(x) ,
 \tag{26}$$

where  $V_j$  is a deterministic function,  $\lambda_j$  a stochastic function and  $K$  the number of modes of the decomposition. Thus, the GSD allows to compute a solution where the deterministic and the stochastic basis functions are not fixed a priori. The flexible basis functions allow to find a solution, which has significant less modes, i.e.  $K \ll N$ , but has nearly the same approximation quality.

In [14] it is shown how to achieve the modes of an optimal approximation in the energy norm  $\|\cdot\|_A$  of the problem, i.e. such that

$$\left\| u - \sum_{j=1}^K \lambda_j V_j \right\|_A^2 = \min_{\lambda, U} \left\| u - \sum_{j=1}^K \lambda_j V_j \right\|_A .
 \tag{27}$$

Details about the GSD method, proofs for the optimality of the approximation, implementation details and numerical tests can be found in [14] and in the supplementary material of this contribution. In our implementation the power-type GSD presented in [14] is used.

## 4 Results

In the following we demonstrate the performance of our stochastic segmentation approach. Our first input image data set consists of  $M = 5$  samples from the artificial "street sequence" [11], the second dataset consists of  $M = 45$  sample images from ultrasound (US) imaging of a structure in the forearm, acquired within 2 seconds. Note that we do not consider the street sequence as an image sequence here, instead we use 5 consecutive frames as samples of the noisy and uncertain acquisition of the same object. From the samples we compute the gPC representation using  $n = 10$  (US), respectively  $n = 4$  (street scene) RVs with the method described in Section 2.3. Our images have a resolution of  $100 \times 100$  pixels. We use a maximal polynomial degree of  $p = 3$  leading to a gPC dimension  $N = 286$  (US) and  $N = 35$  (street scene), respectively. For the reduction of the complexity by the GSD we set  $K = 6$ . Furthermore, we use  $\nu = 0.00075$  and  $k_\varepsilon = 2.0h$  in all computations, where  $h$  is the grid spacing. To show the influence of the RVs, we have also used the US data using the mean value only ( $n = 0$ ).

### 4.1 Street Image Data Set

Between the samples of the street sequence the camera position and the position of the car differs, thus the edge detection using (17) should show a high variance at edges close to the camera (thus moving much) and around the moving car. The results depicted in Fig. 2 match with these expectations. Indeed, in the

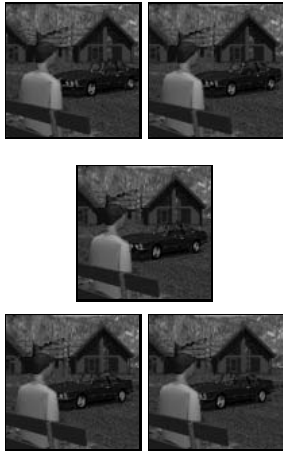

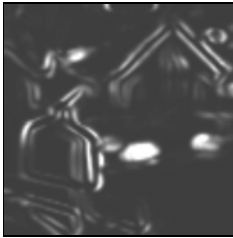

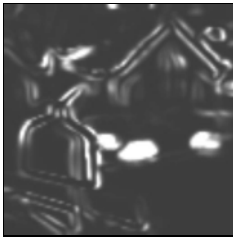
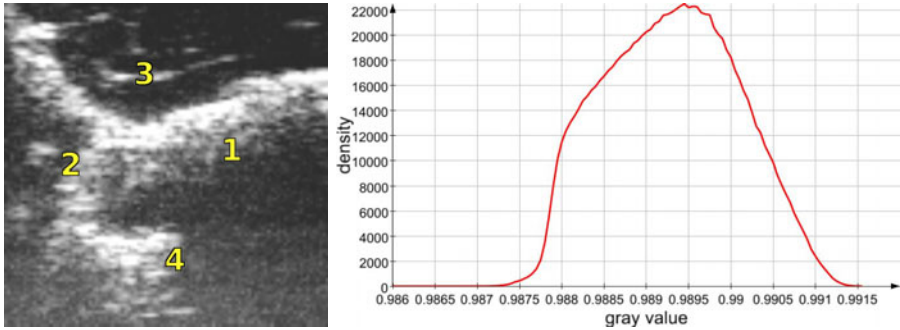
Samples		$\mathbb{E}(\phi)$	$\text{Var}(\phi)$
	GSD		
	MonteCarlo		

Fig. 2. Results of the segmentation of the street scene



**Fig. 3.** Left: Denotation of image regions. The image shows a structure in the forearm. Right: Probability density function of a single pixel from the resulting phase field.


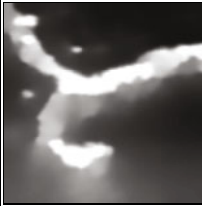

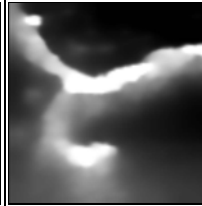

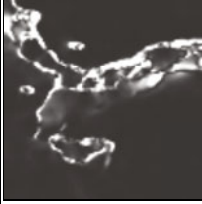
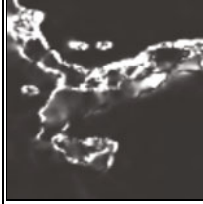





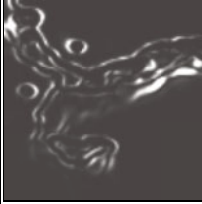
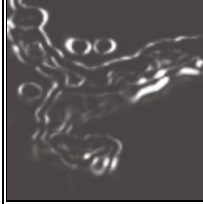
**Table 1.** Comparison of the computation times of different methods for the discretization of SPDEs (measured for the street scene)

Numerical method	Computation time	Number of samples
MonteCarlo	about 35 hours	10000
Stochastic Collocation	about 7 hours	about 2000
GSD	about 2 hours	n/a

region around the wheels of the car and around the right shoulder of the person the edge detection is most influenced by the moving camera, respectively the varying gray values between the samples at the edges. Also around the edges in the background the variance is increased due to the moving camera. A comparison of the results of our GSD implementation with a simple Monte Carlo method with 10000 sample computations shown in Fig. 2 reveals that both approaches lead to similar results. In Table 1 we report the execution times on a typical desktop PC for the Monte Carlo sampling, a more sophisticated sampling using stochastic collocation [19] and the GSD method discussed here. We see that the GSD implementation is about 20 times faster as the classical sampling, however the intrusive GSD needs more implementational effort than the non-intrusive sampling techniques, which can reuse existing deterministic code, because sampling techniques solve the classical Ambrosio-Tortorelli model for every sample and compute stochastic quantities like the variance afterwards from the results on the samples.

## 4.2 Ultrasound Samples

The conversion of the input samples into the gPC expansion as described in Section 2.3 leads to the representation of the stochastic ultrasound image in a 286-dimensional space. Thus, the only meaningful way of visualizing this stochastic image is via stochastic moments like mean and variance. Fig. 4 shows the mean

		10 RVs			mean only
		$\varepsilon = 0.2h, \mu = \frac{1}{300}$	$\varepsilon = 0.4h, \mu = \frac{1}{300}$	$\varepsilon = 0.4h, \mu = \frac{1}{400}$	$\varepsilon = 0.4h, \mu = \frac{1}{400}$
Image	E				
	Var				n/a
Phase Field	E				
	Var				n/a

**Fig. 4.** The mean and variance of the resulting image and phase field for different parameter settings and numbers of RVs using the ultrasound data

and the variance of the phase field  $\phi$  and the smoothed image  $u$  for different settings of the smoothing coefficient  $\mu$  and the phase field width  $\varepsilon$ . The algorithm needs about 100 iterations, i.e. alternate solutions of (19) for  $u$  and  $\phi$ . However, in the first steps the convergence is very fast and already after about 10 iterations no visible difference in  $u$  and  $\phi$  can be seen (cf. movie in the supplementary material that shows the solution iterations).

From the variance image of the phase field the identification of regions where the input distribution has a strong influence on the segmentation result (areas with high variance) is straight forward. A benefit of our new stochastic edge detection via the phase field  $\phi$  is that it allows for an identification of edges in a way that is robust with regard to parameter changes. Indeed, in particular within the four regions marked in the left picture of Fig. 3 the expectation of

the phase field is highly influenced by the choice of  $\mu$  and  $\nu$  as can be seen in Fig. 4. The blurred edge at position 1 is seen in the expectation of the phase field only when a narrow phase field is used. In region 2 we have a different situation in which the edge can be identified only using a widish phase field. Also the edges at positions 3 and 4 can be identified using adjusted parameters. However, note that in case one of these edges is not seen in the expectation of  $\phi$  because of a particular choice of parameters, a high variance of  $\phi$  indicates the possible existence of an edge. This is in particular obvious for the regions 1 and 2.

Moreover, our algorithm can estimate the reliability of detected edges: A low mean phase field value and a low variance indicate, that the edge is robust and not influenced by the noise and uncertainty of the acquisition process. This is for example true for the edges on the top of the structure shown here. In contrast to that a high variance in regions with a high or low mean phase field value (e.g. the labeled regions 1-4) indicates regions, where the detected edge is highly sensitive to the noise and uncertain acquisition process.

Also, we can easily extract the distribution of the gray values for any pixel location inside the image and the phase field from the gPC expansion obtained via GSD. In Fig. 3, right, we show the probability density function of a pixel from the phase field computed via the GSD.

## 5 Conclusions

We have presented an extension of the well known Ambrosio-Tortorelli phase field approximation of the Mumford-Shah functional to stochastic images. Our approach allows us to propagate information about the distribution of the gray values in the input image, which result from noise or erroneous measurements, through the segmentation process, leading to a segmentation result that contains information about the reliability of the segmentation. The resulting SPDEs are discretized by the generalized polynomial chaos approach and a generalized spectral decomposition method. We have shown the application of the segmentation to artificial sample images as well as to noisy ultrasound image samples. In an ongoing work we investigate the use of our algorithm on the basis of noise models instead of multiple input image samples.

In particular for medical applications of quantitative image processing we envisage that our approach can be a basis for superior results, since it allows to measure the size of lesions including reliability estimates. But also other applications, e.g. material science, quality control, geography etc. can benefit from the reliability estimates. In the future we plan to investigate a stochastic extension of edge linking methods [7] for the Mumford-Shah functional. Also, we will study stochastic extensions of sharp interface segmentation methods like level set based approaches.

**Acknowledgements.** We acknowledge R.M. Kirby from the University of Utah, USA for fruitful discussions and D. Ojdanic from Fraunhofer MEVIS, Bremen, Germany for providing the ultrasound data set.

## References

1. Ambrosio, L., Tortorelli, M.: Approximation of functionals depending on jumps by elliptic functionals via gamma-convergence. *Comm. Pure Appl. Math.* 43(8), 999–1036 (1990)
2. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *Int. J. of Computer Vision* 61(3), 211–231 (2005)
3. Cameron, R.H., Martin, W.T.: The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals. *The Annals of Mathematics* 48(2), 385–392 (1947)
4. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. *Int. J. of Computer Vision* 22(1), 61–79 (1997)
5. Chan, T., Vese, L.: Active contours without edges. *IEEE T. Image. Process.* 10(2), 266–277 (2001)
6. Debusschere, B.J., Najm, H.N., Pébay, P.P., Knio, O.M., Ghanem, R.G., Le Maître, O.P.: Numerical challenges in the use of polynomial chaos representations for stochastic processes. *SIAM J. Sci. Comput.* 26(2), 698–719 (2005)
7. Erdem, E., Sancar Yilmaz, A., Tari, S.: Mumford-Shah regularizer with spatial coherence. In: Sgallari, F., Murli, A., Paragios, N. (eds.) *SSVM 2007. LNCS*, vol. 4485, pp. 545–555. Springer, Heidelberg (2007)
8. Ghanem, R.G., Spanos, P.D.: *Stochastic finite elements: A spectral approach*. Springer-Verlag, New York (1991)
9. Loeve, M.: *Probability theory*, 4th edn. Springer, New York (1977)
10. Malladi, R., Sethian, J.A., Vemuri, B.C.: Evolutionary fronts for topology-independent shape modeling and recovery. In: Eklundh, J.-O. (ed.) *ECCV 1994, Part I. LNCS*, vol. 801, pp. 3–13. Springer, Heidelberg (1994)
11. McCane, B., Novins, K., Crannitch, D., Galvin, B.: On benchmarking optical flow. *Comput. Vis. Image Underst.* 84(1), 126–143 (2001)
12. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.* 42(5), 577–685 (1989)
13. Nestares, O., Fleet, D.J., Heeger, D.J.: Likelihood functions and confidence bounds for total-least-squares problems. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 523–530 (2000)
14. Nouy, A.: A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations. *Computer Methods in Applied Mechanics and Engineering* 196(45–48), 4521–4537 (2007)
15. Preusser, T., Scharr, H., Krajssek, K., Kirby, R.: Building blocks for computer vision with stochastic partial differential equations. *Int. J. of Computer Vision* 80(3), 375–405 (2008)
16. Stefanou, G., Nouy, A., Clement, A.: Identification of random shapes from images through polynomial chaos expansion of random level-set functions. *Int. J. for Numerical Methods in Engineering* 79(2), 127–155 (2009)
17. Weber, J., Malik, J.: Robust computation of optical flow in a multi-scale differential framework. *Int. J. of Computer Vision* 14(1), 67–81 (1995)
18. Wiener, N.: The homogeneous chaos. *Am. J. of Mathematics* 60(4), 897–936 (1938)
19. Xiu, D., Hesthaven, J.S.: High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.* 27(3), 1118–1139 (2005)
20. Xiu, D., Karniadakis, G.E.: The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* 24(2), 619–644 (2002)



# Multiple Hypothesis Video Segmentation from Superpixel Flows

Amelio Vazquez-Reina<sup>1,2</sup>, Shai Avidan<sup>3</sup>, Hanspeter Pfister<sup>1</sup>, and Eric Miller<sup>2</sup>

<sup>1</sup> School of Engineering and Applied Sciences, Harvard University, MA, USA

<sup>2</sup> Department of Computer Science, Tufts University, MA, USA

<sup>3</sup> Adobe Systems Inc., USA

**Abstract.** Multiple Hypothesis Video Segmentation (MHVS) is a method for the unsupervised photometric segmentation of video sequences. MHVS segments arbitrarily long video streams by considering only a few frames at a time, and handles the automatic creation, continuation and termination of labels with no user initialization or supervision. The process begins by generating several pre-segmentations per frame and enumerating multiple possible trajectories of pixel regions within a short time window. After assigning each trajectory a score, we let the trajectories compete with each other to segment the sequence. We determine the solution of this segmentation problem as the MAP labeling of a higher-order random field. This framework allows MHVS to achieve spatial and temporal long-range label consistency while operating in an on-line manner. We test MHVS on several videos of natural scenes with arbitrary camera and object motion.

## 1 Introduction

Unsupervised photometric video segmentation, namely the automatic labeling of a video based on texture, color and/or motion, is an important computer vision problem with applications in areas such as activity recognition, video analytics, summarization, surveillance and browsing [1,2]. However, despite its significance, the problem remains largely open for several reasons.

First, the unsupervised segmentation of arbitrarily long videos requires the automatic creation, continuation and termination of labels to handle the free flow of objects entering and leaving the scene. Due to occlusions, objects often merge and split in multiple 2D regions throughout a video. Such events are common when dealing with natural videos with arbitrary camera and object motion. A complete solution to the problem of multiple-object video segmentation requires tracking object fragments and handling splitting or merging events.

Second, robust unsupervised video segmentation must take into account spatial and temporal long-range relationships between pixels that can be several frames apart. Segmentation methods that track objects by propagating solutions frame-to-frame [3,4] are prone to overlook pixel relationships that span several frames.



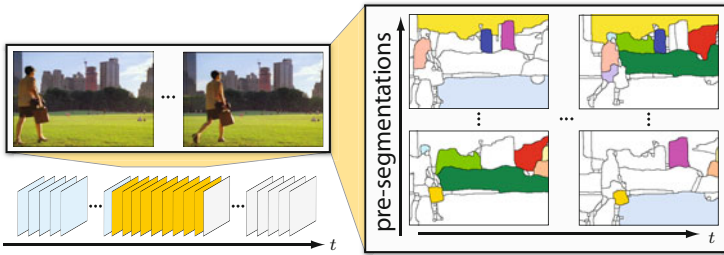
**Fig. 1.** Results from the on-line, unsupervised, photometric segmentation of a video sequence with MHVS. **Top:** original frames. **Bottom:** segmented frames. MHVS keeps track of multiple possible segmentations, collecting evidence across several frames before assigning a label to every pixel in the sequence. It also automatically creates and terminates labels depending on the scene complexity and as the video is processed.

Finally, without knowledge about the number of objects to extract from an image sequence, the problem of unsupervised video segmentation becomes strongly ill-posed [5]. Determining the optimal number of clusters is a fundamental problem in unsupervised data clustering [5].

**Contributions.** MHVS is, to the best of our knowledge, the first solution to the problem of fully unsupervised on-line video segmentation that can effectively handle arbitrarily long sequences, create and terminate labels as the video is processed, and still preserve the photometric consistency of the segmentation across several frames.

Although the connections between tracking and video segmentation are well discussed in *e.g.* [6, 3, 7, 4, 8], we present the first extension of the idea of deferred inference from Multiple Hypothesis Tracking (MHT) [9, 10] to the problem of unsupervised, multi-label, on-line video segmentation. MHVS relies on the use of space-time segmentation hypotheses, corresponding to alternative ways of grouping pixels in the video. This allows MHVS to postpone segmentation decisions until evidence has been collected across several frames, and to therefore operate in an on-line manner while still considering pixel relationships that span multiple frames. This extension offers other important advantages. Most notably, MHVS can dynamically handle the automatic creation, continuation and termination of labels depending on the scene complexity, and as the video is processed.

We also show how higher-order conditional random fields (CRFs), which we use to solve the hypothesis competition problem, can be applied to the problem of unsupervised on-line video segmentation. Here, we address two important challenges. First, the fact that only a subset of the data is available at any time



**Fig. 2.** **Left:** MHVS labels a video stream in an on-line manner considering several frames at a time. **Right:** For each processing window, MHVS generates multiple pre-segmentations per frame, and finds sequences of superpixels (shown as colored regions) that match consistently in time. Each of these sequences, called a superpixel flow, is ranked depending on its photometric consistency and considered as a possible label for segmentation. The processing windows overlap one or more frames to allow labels to propagate from one temporal window to the next.

during the processing, and second, that the labels themselves must be inferred from the data. A working example of MHVS is illustrated on Fig. 1.

**Previous work.** Some of the common features and limitations found in previous work on video segmentation include:

1. The requirement that all frames are available at processing time and can be segmented together [6,11,12,13]. While this assumption holds for certain applications, the segmentation of arbitrarily long video sequences requires the ability to segment and track results in a continuous, sequential manner (we refer to this as *on-line* video segmentation). Unfortunately, those methods that can segment video in an on-line manner usually track labels from frame to frame [3,7,4] (*i.e.*, they only consider two frames at a time), which makes them sensitive to segmentation errors that gradually accumulate over time.
2. The user is often required to provide graphical input in the form of scribbles, seeds, or even accurate boundary descriptions in one or multiple frames to initiate or facilitate the segmentation [14,11]. This can be helpful or even necessary for the high level grouping of segments or pixels, but we aim for an automatic method.
3. The assumption that the number of labels is known *a priori* or is constant across frames [15,16,17,18,12,14] is useful in some cases such as foreground-background video segmentation [18,12,14], but only a few methods can adaptively and dynamically determine the number of labels required to photometrically segment the video. Such ability to adjust is especially important in on-line video segmentation, since the composition of the scene tends to change over time.

Recently, Brendel and Todorovic [6] presented a method for unsupervised photometric video segmentation based on mean-shift and graph relaxation. The main difference between their work and MHVS is that our method can operate in an on-line manner and consider multiple segmentation hypotheses before segmenting the video stream.

## 2 An Overview of MHVS

The three main steps in MHVS are: hypotheses enumeration, hypotheses scoring, and hypotheses competition.

A *hypothesis* refers to one possible way of grouping several pixels in a video, *i.e.*, a correspondence of pixels across multiple frames. More specifically, we define a hypothesis as a grouping or flow of *superpixels*, where a superpixel refers to a contiguous region of pixels obtained from a tessellation of the image plane without overlaps or gaps. This way, each hypothesis can be viewed as a possible label that can be assigned to a group of pixels in a video (see Fig. 2).

Since different hypotheses represent alternative trajectories of superpixels, hypotheses will be said to be *incompatible* when they overlap; that is, when one or more pixels are contained in more than one hypothesis. In order to obtain a consistent labeling of the sequence, we aim for the exclusive selection of only one hypothesis for every set of overlapping hypotheses (see an example in Fig. 3).

Depending on the photometric consistency of each hypothesis, we assign them a score (a likelihood). This allows us to rank hypotheses and compare them in probabilistic terms. The problem of enumeration and scoring of hypotheses is discussed in Section 3. Once hypotheses have been enumerated and assigned a score, we make them compete with each other to label the video sequence. This competition penalizes the non-exclusive selection between hypotheses that are incompatible in the labeling. In order to resolve the hypotheses competition problem, MHVS relies on MAP estimation on a higher-order conditional random field (CRF). In this probabilistic formulation, hypotheses will be considered as labels or classes that can be assigned to superpixels on a video. Details about this step are covered in Section 4.

For the segmentation of arbitrarily long video sequences, the above process of hypotheses enumeration, scoring and competition is repeated every few frames using a sliding window. By enumerating hypotheses that include the labels from the segmentation of preceding windows, solutions can be propagated sequentially throughout an arbitrarily long video stream.

## 3 Enumeration and Scoring of Hypotheses

The enumeration of hypotheses is a crucial step in MHVS. Since the number of all possible space-time hypotheses grows factorially with frame resolution and video length, this enumeration must be selective. The pruning or selective sampling of



**Fig. 3.** Two hypotheses that are incompatible. The hypotheses (shown in green and red) overlap on the first two frames. The segmentation of the sequence should ensure their exclusive selection. MHVS ranks hypotheses photometrically and penalizes the non-consistent selection of the most coherent ones over time.

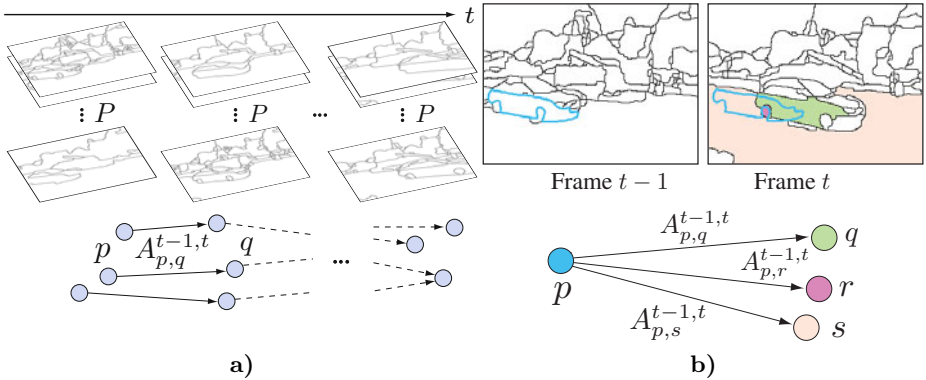
hypotheses is a common step in the MHT literature, and it is usually solved via a “gating” procedure [19].

We address the enumeration and scoring of hypotheses in two steps. First, we generate multiple pre-segmentations for each frame within the processing window using segmentation methods from the literature, *e.g.*, [20], [21]. Then, we match the resulting segments across the sequence based on their photometric similarity. Those segments that match consistently within the sequence will be considered as hypotheses (possible labels) for segmentation.

The above approach can be modeled with a Markov chain of length equal to that of the processing window. This allows us to look at hypotheses as time sequences of superpixels that are generated by the chain, with the score of each hypothesis given by the probability of having the sequence generated by the chain.

We formalize this approach as follows. Given a window of  $F$  consecutive frames from a video stream, we build a weighted, directed acyclic graph  $G = (V, E)$  that we denote as a *superpixel adjacency graph*. In this graph, a node represents a superpixel from one of the pre-segmentations on some frame within the processing window, and an edge captures the similarity between two temporally adjacent superpixels (superpixels that overlap spatially but belong to two different and consecutive frames). Edges are defined to point from a superpixel from one of the pre-segmentations on time  $t$  to a superpixel from one of the pre-segmentations on  $t + 1$ . Fig. 4 shows an illustration of how this graph is built.

The above graph can be thought as the transition diagram of a Markov chain of length  $F$  [22]. In this model, each frame is associated with a variable that represents the selection of one superpixel in the frame, and the transition probabilities between two variables are given by the photometric similarity between two temporally adjacent superpixels. By sampling from the chain, for example, via ancestral sampling [22] or by computing shortest paths in the transition diagram, we can generate hypotheses with strong spatio-temporal coherency.



**Fig. 4.** Construction of the superpixel adjacency graph for the enumeration of hypotheses (flows of superpixels). **(a)** For each processing window, MHVS generates  $P$  pre-segmentations on each frame. Each of them groups pixels at different scales and according to different photometric criteria. The nodes in the graph represent superpixels from some of the pre-segmentations on each frame, and the edges capture the photometric similarity between two temporally adjacency superpixels. **(b)** Two superpixels are considered to be temporally adjacent if they overlap spatially but belong to two different and consecutive frames.

More specifically, for a given window of  $F$  frames, and the set of all superpixels  $\mathcal{V} = \{V_1, \dots, V_F\}$  generated from  $P$  pre-segmentations on each frame, we can estimate the joint distribution of a sequence of superpixels  $(\mathbf{z}_1, \dots, \mathbf{z}_F)$  as

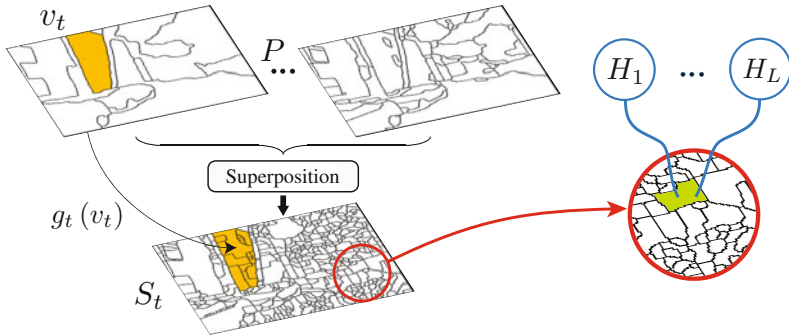
$$p(\mathbf{z}_1, \dots, \mathbf{z}_F) = p(\mathbf{z}_1) \cdot \prod_{t=2}^{t=F} A_{j,k}^{t-1,t}, \quad (1)$$

where the transition matrices  $A_{j,k}^{t-1,t}$  capture the photometric similarity between two temporally adjacent superpixels  $\mathbf{z}_{t-1} = j$  and  $\mathbf{z}_t = k$ , and are computed from the color difference between two superpixels in LUV colorspace, as suggested in [23]. In order to generate hypotheses that can equally start from any superpixel on the first frame, we model the marginal distribution of the node  $\mathbf{z}_1$  as a uniform distribution. Further details about the generation of pre-segmentations and the sampling from the Markov chain are discussed in Section 5.

Once a set of hypotheses has been enumerated, we measure their temporal coherency using the joint distribution of the Markov chain. Given a set of  $L$  hypotheses  $\mathcal{H} = \{H_1, \dots, H_L\}$ , we define the score function  $s: \mathcal{H} \rightarrow [0, 1]$  as:

$$s(H_k) = N_1 \cdot p(\mathbf{z}_1 = v_1, \dots, \mathbf{z}_F = v_F) = \prod_{t=2}^F A_{v_{t-1}, v_t}^{t-1,t}, \quad (2)$$

where  $(v_1, \dots, v_F)$  is a sequence of superpixels comprising a hypothesis  $H_k$  and  $N_1$  is the number of superpixels on the first frame.



**Fig. 5.** We define our higher-order conditional random field on a sequence of fine grids of superpixels  $\mathcal{S} = \{S_1, \dots, S_F\}$ . Each grid  $S_t$  is obtained as the superposition of the  $P$  tessellations that were generated for the enumeration of hypotheses. The mapping  $g_t$  takes superpixels  $v_t$  from one of the pre-segmentations to the superposition  $S_t$ . Each superpixel in  $S_t$  is represented in our CRF with a random variable that can be labeled with one of the hypotheses  $\{H_1, \dots, H_L\}$ .

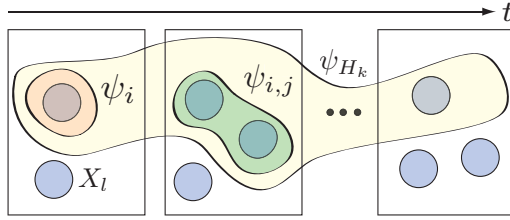
**Propagation of solutions.** The above approach needs to be extended to also enumerate hypotheses that propagate the segmentation results from preceding processing windows. We address this problem by allowing our processing windows to overlap one or more frames. The overlap can be used to consider the superpixels resulting from the segmentation of each window when enumerating hypothesis in the next window. That is, the set of pre-segmented superpixels  $\mathcal{V} = \{V_1, \dots, V_F\}$  in a window  $w$ ,  $w > 1$ , is extended to include the superpixels that result from the segmentation of the window  $w - 1$ .

## 4 Hypotheses Competition

Once hypotheses have been enumerated and scored for a particular window of frames, we make them compete with each other to label the sequence. We determine the solution to this segmentation problem as the MAP labeling of a random field defined on a sequence of fine grids of superpixels. This framework allows us to look at hypotheses as labels that can be assigned to random variables, each one representing a different superpixel in the sequence (see Fig. 5).

Our objective function consists of three terms. A unary term that measures how much a superpixel within the CRF grid agrees with a given hypothesis, a binary term that encourages photometrically similar and spatially neighboring superpixels to select the same hypothesis, and a higher-order term that forces the consistent labeling of the sequence with the most photometrically coherent hypotheses over time (See Fig. 6 for an illustration).

We formalize this as follows. For each processing window of  $F$  frames, we define a random field of  $N$  variables  $X_i$  defined on a sequence of grids of superpixels  $\mathcal{S} = \{S_1, \dots, S_F\}$ , one for each frame. Each grid  $S_t$  is obtained as the superposition of the  $P$  pre-segmentations used for the enumeration of hypotheses, and



**Fig. 6.** The unary, pairwise and higher-order potentials,  $\psi_i$ ,  $\psi_{i,j}$  and  $\psi_{H_k}$ , respectively, control the statistical dependency between random variables  $X_i$ , each one representing a different superpixel within the processing window.

yields a mapping  $g_t$  that takes every superpixel from the pre-segmentations to the set  $S_t$  (see Fig. 5). The random variables  $X_i$  are associated with superpixels from  $S$ , and take values from the label set  $\mathcal{H} = \{H_1, \dots, H_L\}$ , where each hypothesis  $H_k$  is sampled from the Markov chain described in the previous section.

A sample  $\mathbf{x} = (x_1, \dots, x_N) \in \mathcal{H}^N$  from the field, *i.e.* an assignment of labels (hypotheses) to its random variables, is referred to as a *labeling*. From the Markov-Gibbs equivalence, the MAP labeling  $\mathbf{x}^*$  of the random field takes the form:

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathcal{H}^N} \sum_{c \in \mathcal{C}} \alpha_c \psi_c(\mathbf{x}_c), \quad (3)$$

where the potential functions  $\psi_c$  are defined on cliques of variables  $c$  from some set  $\mathcal{C}$ , and  $\alpha_c$  are weighting parameters between the different potentials. The labeling  $\mathbf{x}_c$  represents the assignment of the random variables  $X_i$  within the clique  $c$  to their corresponding values in  $\mathbf{x}$ .

We next define three different types of potentials  $\psi_c$  (representing penalties on the labeling) for our objective function in Eq. 3. The potentials enforce the consistent photometric labeling of the sequence. The unary potentials favor the selection of hypotheses that provide a high detail (fine) labeling of each frame. The pairwise potentials encourage nearby superpixels to get the same label, depending on their photometric similarity. Finally, the higher-order potentials force the exclusive selection of hypotheses that are incompatible with each other.

**Unary potentials.** The mappings  $g = (g_1, \dots, g_F)$  between the pre-segmentations and the grids  $S_t$  (see Fig. 5) are used to define the penalty of assigning a hypothesis  $x_i$  to the random variable  $X_i$  representing the superpixel  $s_i$  as

$$\psi_i(x_i) = 1 - d(s_i, g(x_i)), \quad (4)$$

where  $g(x_i)$  represents the mapping of the superpixels within the hypothesis  $x_i$  to the set of superpixels  $S$ . The function  $d(a, b)$  measures the Dice coefficient  $\in [0, 1]$  on the plane between the sets of pixels  $a$  and  $b$  (the spatial overlap between  $a$  and  $b$ ), and is defined as  $d(a, b) = 2|a \cap b| / (|a| + |b|)$ . Since the set of superpixels  $\{S_1, \dots, S_F\}$  represents an over-segmentation on each frame (it is obtained from a superposition of tessellations), the unary potential favors



labelings of the sequence with spatially thin hypotheses, *i.e.* those with the highest overlap with superpixels on the CRF grid, in the Dice-metric sense.

**Pairwise potentials.** We define the following potential for every pair of spatially adjacent superpixels  $s_i, s_j$  in each frame:

$$\psi_{i,j}(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j \\ b(i, j) & \text{otherwise,} \end{cases} \quad (5)$$

where  $b(i, j)$  captures the photometric similarity between adjacent superpixels, and can be obtained by sampling from a boundary map of the image. The above potential guarantees a discontinuity-preserving labeling of the video, and penalizes label disagreement between neighboring superpixels that are photometrically similar [24]. A discussion on the choice of  $b(i, j)$  is given in Section 5.

**Higher-order potentials.** As mentioned in Section 2, we penalize the non-exclusive selection of hypotheses that are incompatible with each other. To do this, we design a higher-order potential that favors the consistent selection of the most photometrically coherent hypotheses over time. The notion of label consistency was formalized by Kohli *et al.* in [25] and [26] with the introduction of the Robust  $P^n$  model, which they applied to the problem of supervised multi-class image segmentation. Here, we use this model to penalize label disagreement between superpixels comprising hypotheses of high photometric coherency. For each hypothesis  $H_k$ , we define the following potential:

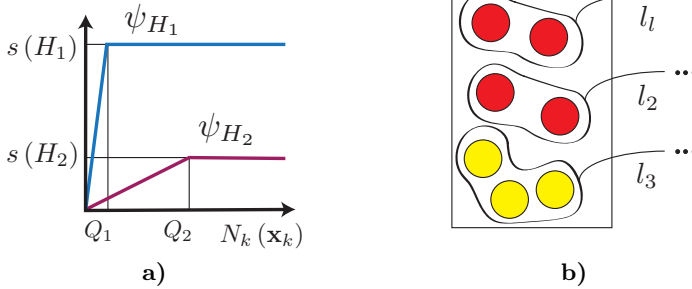
$$\psi_{H_k}(\mathbf{x}_k) = \begin{cases} N_k(\mathbf{x}_k) \frac{1}{Q_k} s(H_k) & \text{if } N_k(\mathbf{x}_k) \leq Q_k \\ s(H_k) & \text{otherwise,} \end{cases} \quad (6)$$

where  $\mathbf{x}_k$  represents the labeling of the superpixels comprising the hypothesis  $H_k$ , and  $N_k(\mathbf{x}_k)$  denotes the number of variables not taking the dominant label (*i.e.*, it measures the label disagreement within the hypothesis). The score function  $s(H_k)$  defined in the previous section measures the photometric coherency of the hypothesis  $H_k$  (see Eq. 2). The truncation parameter  $Q_k$  controls the rigidity of the higher-order potential [25], and we define it as:

$$Q_k = \frac{1 - s(H_k)}{\max_{m \in [1, L]} (1 - s(H_m))} \cdot \frac{|c|}{2}. \quad (7)$$

The potential  $\psi_{H_k}$  with the above truncation parameter gives higher penalties to those labelings where there is strong label disagreement between superpixels that belong to highly photometrically coherent hypotheses (the more photometrically coherent a hypothesis is, the higher the penalty for disagreement between the labels of the CRF superpixels comprising it). See Fig. 7(a) for an example.

**Labeling.** Once we have defined unary, binary and higher-order potentials for our objective function in Eq. 3, we approximate the MAP estimate of the CRF using a graph cuts solver for the Robust  $P^n$  model [25]. This solver relies on a



**Fig. 7.** (a) Higher-order penalty ( $y$ -axis) as a function of label disagreement within a hypothesis ( $x$ -axis) for two overlapping hypotheses  $H_1$  and  $H_2$ , with  $H_1$  being more photometrically coherent than  $H_2$ . The potential  $\psi_{H_1}$  strongly penalizes any label disagreement within  $H_1$ , while  $\psi_{H_2}$  tolerates significantly higher label disagreement within  $H_2$ . (b) The colored circles represent superpixels that were labeled in the preceding processing window (each color being a different label). The groupings  $l_1, l_2$  and  $l_3$  are the result of the MAP labeling within the current processing window. Depending on the selection of  $\gamma_1$  and  $\gamma_2$  (see text),  $l_1$  and  $l_2$  are considered as new labels or mapped to the label depicted in red.

sequence of alpha-expansion moves that are binary, quadratic and submodular, and therefore exactly computable in polynomial time [25]. From the association between variables  $X_i$  and the superpixels in  $S$ , this MAP estimate also yields the segmentation of all the pixels within the processing window.

**Handling mergers and splits.** The implicit (non-parametric) object boundary representation provided by the random field [24] allows MHVS to easily handle merging and splitting of labels over time; when an object is split, the MAP labeling of the graph yields disconnected regions that share the same label. Since labels are propagated across processing windows, when the parts come back in contact, the labeling yields a single connected region with the same label. The automatic merging of object parts that were not previously split in the video is also implicitly handled by MHVS. This merging occurs when the parts of an object are included within the same hypothesis (i.e. one of the pre-segmentations groups the parts together).

In order to create new labels for parts of old labels, when the parts become distinguishable enough over time to be tracked, a final mapping of labels is done before moving to the next processing window. We handle this scenario by comparing the spatial overlap between new labels (from the current processing window) and old labels (from the preceding processing window). We check for new labels  $l$  that significantly overlap spatially with some old label  $p$ , but barely overlap with any other old label  $q$ . We can measure such overlaps using their Dice coefficients, and we denote them by  $\gamma_p$  and  $\gamma_q$ . Then, if  $\gamma_p > \gamma_1$  and  $\gamma_q < \gamma_2$ ,  $\forall q \neq p$ , for a pair of fixed parameters  $\gamma_1, \gamma_2 \in [0, 1]$ , we map the label  $l$  to  $p$ , otherwise  $l$  is considered a new label (see Fig. 7(b) for an example).

## 5 Experimental Results

Most previous work on unsupervised photometric video segmentation has focused on the segmentation of sequences with relatively static backgrounds and scene complexity [4,6,12,16]. In this paper, however, we show results from applying MHVS to natural videos with arbitrary motion on outdoor scenes. Since existing datasets of manually-labeled video sequences are relatively short (often less than 30 frames), and usually contain a few number of labeled objects (often only foreground and background), we collected five videos of outdoor scenes with 100 frames each, and manually annotated an average of 25 objects per video every three frames. The videos include occlusions, objects that often enter and leave the scene, and dynamic backgrounds (see Figs. 1 and 8 for frame examples).

We compared MHVS with spatio-temporal mean-shift (an *off-line* method, similar to [13]), and pairwise graph propagation (an *on-line* method with frame-to-frame propagation, similar to [4]). In both methods we included color, texture and motion features. For the test with mean-shift, each video was processed in a single memory-intensive batch. For our MHVS tests,  $F$  was set to 5 frames to meet memory constraints, but values between 3 and 10 gave good results in general. The size of the processing window was also observed to balance MHVS’s ability to deal with strong motion while preserving long-term label consistency. We used an overlap of one frame between processing windows and generated  $P = 30$  pre-segmentations per frame using the *gPb* boundary detector introduced by Maire *et al.* [21], combined with the OWT-UCM algorithm from [27].

As mentioned in Section 3, hypotheses can be obtained via ancestral sampling [22] (*i.e.* sampling from the conditional multinomial distributions in the topological order of the chain), or by computing shortest paths in the transition diagram from each superpixel on the first frame to the last frame in the window (*i.e.* computing the most likely sequences that start with each value of the first variable in the chain). We follow this second approach. Neither guarantees that every CRF superpixel is visited by a hypothesis. In our implementation, such CRF superpixels opt for a dummy (void) label, and those that overlap with the next processing window are later considered as sources for hypotheses. The parameters  $\alpha_e$  weighting the relative importance between the unary, pairwise and higher-order potentials in Eq. 3 were set to 10, 2 and 55, respectively, although similar results were obtained within a 25% deviation from these values. The pairwise difference between superpixels  $b(i, j)$  was sampled from the boundary map generated by OWT-UCM and the parameters  $\gamma_1$  and  $\gamma_2$  that control the mapping of new labels to old labels were set to 0.8 and 0.2, respectively.

We measured the quality of the segmentations using the notion of *segmentation covering* introduced by Arbeláez *et al.* in [27]. The covering of a human segmentation  $S$  by a machine segmentation  $S'$ , can be defined as:

$$C(S' \rightarrow S) = \frac{1}{N} \sum_{V \in S} |V| \cdot \max_{V' \in S'} d(V, V') \quad (8)$$

where  $N$  denotes the total number of pixels in the video, and  $d(V, V')$  is the Dice coefficient in 3D between the labeled spatio-temporal volumes  $V$  and  $V'$



**Fig. 8. Top to fourth row:** Results from the on-line, unsupervised, photometric segmentation of four video sequences of varying degrees of complexity with MHVS. The examples show MHVS's ability to adjust to changes in the scene, creating and terminating labels as objects enter and leave the field of view. **Fourth and fifth row:** Comparison between MHVS (fourth row) and pairwise graph propagation (similar to [4]) (fifth row). The frames displayed are separated by 5-10 frames within the original segmented sequences.

**Table 1.** Best segmentation covering obtained with MHVS, pairwise graph propagation and mean-shift across five outdoor sequences that were manually annotated. Frame examples from Video 1 are shown in Fig. 1, and from Videos 2 to 5 in Fig. 8, top to bottom. Higher segmentation coverings are better.

Method	Video 1	Video 2	Video 3	Video 4	Video 5
MHVS (multi-frame on-line)	<b>0.62</b>	<b>0.59</b>	<b>0.45</b>	<b>0.54</b>	0.42
Graph propagation (pairwise on-line)	0.49	0.37	0.36	0.39	0.34
Mean-shift (off-line)	0.56	0.39	0.34	0.38	<b>0.44</b>

within  $S$  and  $S'$ , respectively. These volumes can possibly be made of multiple disconnected space-time regions of pixels. Table I shows the values of the best segmentation covering achieved by each method on our five videos.

## 6 Discussion and Future Work

In our tests, we observed that sometimes labels have a short lifespan. We attribute this to the fact that it is difficult to find matching superpixels in pre-segmentations of consecutive frames. The use of multiple pre-segmentations per frame was introduced to alleviate this problem, and further measures, such as the use of “track stitching” methods (*e.g.* see [28]) could help reduce label flickering in future work.

**Running time.** The unary, pairwise and higher-order potentials of Eq. 3 are sparse. Each random variable (representing an over-segmented superpixel) overlaps few other hypotheses. No overlap makes the unary and higher-order terms associated with the hypothesis zero. The pre-segmentations, enumeration of hypotheses and measuring of photometric similarities between superpixels can be parallelized, and each processing window must be segmented (Eq. 3 solved) before moving to the next processing window. With this, in our tests, MHVS run on the order of secs/frame using a Matlab-CPU implementation.

**Conclusions.** MHVS is, to the best of our knowledge, the first solution to the problem of fully unsupervised on-line video segmentation that can segment videos of arbitrary length, with unknown number of objects, and effectively manage object splits and mergers. Our framework is general and can be combined with any image segmentation method for the generation of space-time hypotheses. Alternative scoring functions, to the ones presented here, can also be used for measuring photometric coherency or similarity between superpixels.

We believe our work bridges further the gap between video segmentation and tracking. It also opens the possibility of integrating the problem of on-line video segmentation with problems in other application domains such as event recognition or on-line video editing. Future work could include extensions of MHVS based on on-line learning for dealing with full occlusions and improving overall label consistency.

**Acknowledgments.** This work was supported in part by the NSF Grant No. PHY-0835713. We also thank the generous support from Microsoft Research, NVIDIA, the Initiative in Innovative Computing (IIC) at Harvard University, and Jeff Lichtman from the Harvard Center for Brain Science.

## References

1. Turaga, P., Veeraraghavan, A., Chellappa, R.: From videos to verbs: Mining videos for activities using a cascade of dynamical systems. In: CVPR 2007 (2007)
2. Pritch, Y., Rav-Acha, A., Peleg, S.: Nonchronological video synopsis and indexing. PAMI 30, 1971–1984 (2008)
3. Ren, X., Malik, J.: Tracking as repeated figure/ground segmentation. In: CVPR 2007 (2007)

4. Liu, S., Dong, G., Yan, C., Ong, S.: Video segmentation: Propagation, validation and aggregation of a preceding graph. In: CVPR 2008 (2008)
5. Jain, A.: Data clustering: 50 years beyond k-means. *Pattern Recognition Letters* (2009)
6. Brendel, W., Todorovic, S.: Video object segmentation by tracking regions. In: ICCV 2009 (2009)
7. Bugeau, A., Pérez, P.: Track and cut: simultaneous tracking and segmentation of multiple objects with graph cuts. *JIVP*, 1–14 (2008)
8. Yin, Z., Collins, R.: Shape constrained figure-ground segmentation and tracking. In: CVPR 2009 (2009)
9. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. *ACM Comput. Surv.* 38, 13 (2006)
10. Reid, D.B.: An algorithm for tracking multiple targets, vol. 17, pp. 1202–1211 (1978)
11. Wang, J., Xu, Y., Shum, H.Y., Cohen, M.F.: Video tooning. In: SIGGRAPH 2004 (2004)
12. Huang, Y., Liu, Q., Metaxas, D.: Video object segmentation by hypergraph cut. In: CVPR 2009 (2009)
13. De Menthon, D.: Spatio-temporal segmentation of video by hierarchical mean shift analysis. In: SMVP 2002 (2002)
14. Bai, X., Wang, J., Simons, D., Sapiro, G.: Video snapcut: robust video object cutout using localized classifiers. In: SIGGRAPH 2009 (2009)
15. Chan, A., Vasconcelos, N.: Variational layered dynamic textures. In: CVPR 2009 (2009)
16. Hedau, V., Arora, H., Ahuja, N.: Matching images under unstable segmentations. In: CVPR 2008 (2008)
17. Ayvaci, A., Soatto, S.: Motion segmentation with occlusions on the superpixel graph. In: ICCVW 2009 (2009)
18. Unger, M., Mauthner, T., Pock, T., Bischof, H.: Tracking as segmentation of spatial-temporal volumes by anisotropic weighted tv. In: Cremers, D., Boykov, Y., Blake, A., Schmidt, F.R. (eds.) *EMMCVPR 2009*. LNCS, vol. 5681, pp. 193–206. Springer, Heidelberg (2009)
19. Blackman, S., Popoli, R.: *Design and Analysis of Modern Tracking Systems*. Artech House (1999)
20. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. *PAMI* 24, 603–619 (2002)
21. Maire, M., Arbelaez, P., Fowlkes, C., Malik, J.: Using contours to detect and localize junctions in natural images. In: CVPR 2008 (2008)
22. Bishop, C.M.: *Pattern Recognition and Machine Learning*. Springer, Heidelberg (2007)
23. Fulkerson, B., Vedaldi, A., Soatto, S.: Class segmentation and object localization with superpixel neighborhoods. In: ICCV 2009 (2009)
24. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient n-d image segmentation. *IJCV* 70, 109–131 (2006)
25. Kohli, P., Ladický, L., Torr, P.H.S.: Robust higher order potentials for enforcing label consistency. *IJCV* 82, 302–324 (2009)
26. Kohli, P., Kumar, M.P., Torr, P.H.S.: P3 & beyond: Solving energies with higher order cliques. In: CVPR 2007 (2007)
27. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: From contours to regions: An empirical evaluation. In: CVPR 2009 (2009)
28. Ding, T., Sznaiier, M., Camps, O.: Fast track matching and event detection. In: CVPR 2008 (2008)

# Object Segmentation by Long Term Analysis of Point Trajectories\*

Thomas Brox<sup>1,2</sup> and Jitendra Malik<sup>1</sup>

<sup>1</sup> University of California at Berkeley

<sup>2</sup> Albert-Ludwigs-University of Freiburg, Germany  
{brox,malik}@eecs.berkeley.edu

**Abstract.** Unsupervised learning requires a grouping step that defines which data belong together. A natural way of grouping in images is the segmentation of objects or parts of objects. While pure bottom-up segmentation from static cues is well known to be ambiguous at the object level, the story changes as soon as objects move. In this paper, we present a method that uses long term point trajectories based on dense optical flow. Defining pair-wise distances between these trajectories allows to cluster them, which results in temporally consistent segmentations of moving objects in a video shot. In contrast to multi-body factorization, points and even whole objects may appear or disappear during the shot. We provide a benchmark dataset and an evaluation method for this so far uncovered setting.

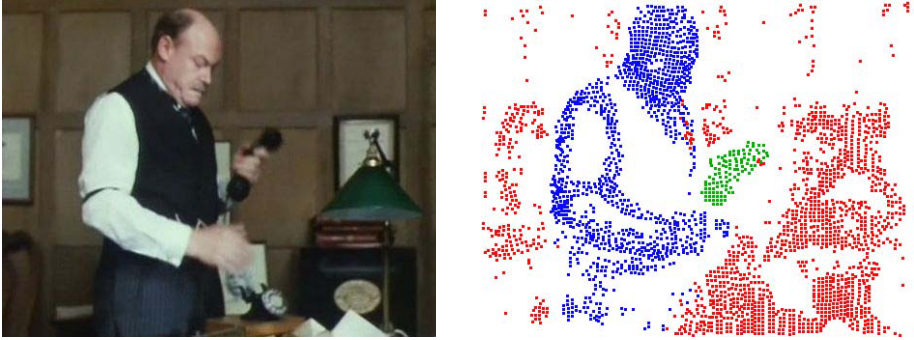
## 1 Introduction

Consider Fig. 1(a). A basic task that one could expect a vision system to accomplish is to detect the person in the image and to infer his shape or maybe other attributes. Contemporary person detectors achieve this goal by learning a classifier and a shape distribution from manually annotated training images. Is this annotation really necessary? Animals or infants are not supplied bounding boxes or segmented training images when they learn to see. Biological vision systems learn objects up to a certain degree of accuracy in an unsupervised way by making use of the natural ordering of the images they see [1]. Knowing that these systems exist, another objective of vision research must be to understand and emulate this capability.

A decisive step towards this goal is object-level segmentation in a purely bottom-up way. This step seems to be impossible given that such segmentation is ambiguous in the very most cases. In Fig. 1 the contrast between the white shirt and the black vest is much higher than the contrast between the vest and the background. How should a bottom-up method know that the shirt and the vest belong to the same object, while the background does not? The missing link

---

\* This work was supported by the German Academic Exchange Service (DAAD) and ONR MURI N00014-06-1-0734.



**Fig. 1. Left: (a)** Bottom-up segmentation from a single input frame is ambiguous. **Right: (b)** Long term motion analysis provides important information for bottom-up object-level segmentation. Only motion information was used to separate the man and even the telephone receiver from the background.

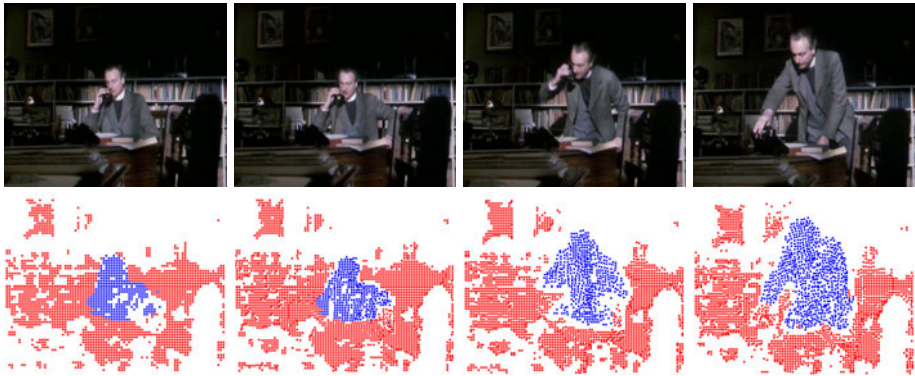
can be established as soon as objects move<sup>1</sup>. Fig. 1 shows a good separation of points on the person versus points on the background with the method proposed in this paper using only motion cues. As these clusters are consistently found for the whole video shot, this provides rich information about the person in various poses.

In this paper we describe a motion clustering method that can potentially be used for unsupervised learning. We argue that temporally consistent clusters over many frames can be obtained best by analyzing long term point trajectories rather than two-frame motion fields. In order to compute such trajectories, we run a tracker we developed in [2], which is based on large displacement optical flow [3]. It provides subpixel accurate tracks on one hand, and can deal with the large motion of limbs or the background on the other. Moreover, in contrast to traditional feature point trackers, it provides arbitrarily dense trajectories, so it allows to assign region labels far more densely. An alternative tracker that will probably work as well with our technique is the one from [4], though the missing treatment of large displacements might be a problem in some sequences.

With these long term point trajectories at hand, we measure differences in how the points move. A key contribution of our method is that we define the distance between trajectories as the maximum difference of their motion over time. The person in Fig. 2 is sitting for a couple of seconds and then rising up. The first part of the shot will not provide any motion information to separate the person from the background. The most valuable cues are available at the point where the person moves fastest. A proper normalization further ensures that scenes with very large motion can be handled the same way as scenes with only little motion.

<sup>1</sup> Potentially even static objects can be separated if there is camera motion. In this paper, however, we consider this case only as a side effect. Generally, active observers will be able to either move themselves or objects of interest in order to generate the necessary motion cues.





**Fig. 2.** Frames 0, 30, 50, 80 of a shot from *Miss Marple: Murder at the vicarage*. Up to frame 30, there is hardly any motion as the person is sitting. Most information is provided when the person is sitting up. This is exploited in the present approach. Due to long term tracking, the grouping information is also available at the first frames.

Given the pairwise distances between trajectories, we can build an affinity matrix for the whole shot and run spectral clustering on this affinity matrix [5,6]. Regarding the task as a single clustering problem, rather than deciding upon a single-frame basis, ensures that trajectories that belong to the same object but did not exist at the same time become connected by the transitivity of the graph. An explicit track repair as in [7] is not needed. Moreover, since we do not assume the number of clusters to be known in advance and the clusters should be spatially compact, we propose a spectral clustering method that includes a spatial regularity constraint allowing for model selection.

In order to facilitate progress in the field of object-level segmentation in videos, we provide an annotated dataset together with an evaluation tool, trajectories, and the binaries of our approach. This will allow for quantitative comparisons in the future. Currently the only reasonably sized dataset with annotation is the Hopkins dataset [8], which is specialized for factorization methods (sparse, manually corrected trajectories, all trajectories have the same length). The new dataset will extend the task to a more general setting where (1) the given trajectories are partially erroneous, (2) occlusion and disocclusion are a frequent phenomenon, (3) shots are generally larger, (4) density plays a role (it will be advantageous to augment the motion cues by static cues), and (5) the number of clusters is not known in advance.

## 2 Related Work

The fact that motion provides important information for grouping is well known and dates back to Koffka and Wertheimer suggesting the Gestalt principle of “common fate” [9]. Various approaches have been proposed for taking this grouping principle into account. Difference images are the most simple way to let

temporally changing structures pop out. They are limited though, as they only indicate a local change but do not provide the reason for that change. This becomes problematic if many or all objects in the scene are subject to a change (e.g. due to a moving camera). Much richer information is provided by optical flow. Numerous motion segmentation methods based on two-frame optical flow have been proposed [10,11,12,13]. The quality of these methods depends on picking a pair of frames with a clear motion difference between the objects. Some works have combined the flow analysis with the learning of an appearance model [14,15]. This leads to temporally consistent layers across multiple frames, but comes along with an increased number of mutually dependent variables. Rather than relying on optical flow, [16] estimates the motion of edges and uses those for a reasoning of motion layers.

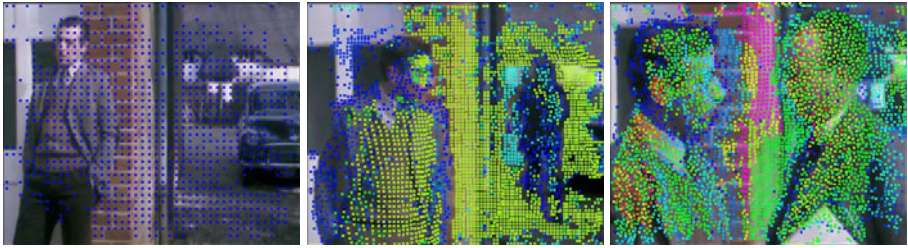
In order to make most use of multiple frames and to obtain temporally consistent segments, a method should analyze trajectories over time. This is nicely exploited by multi-body factorization methods [17,18,19,20]. These methods are particularly well suited to distinguish the 3D motion of rigid objects by exploiting the properties of an affine camera model. On the other hand, they have two drawbacks: (1) factorization is generally quite sensitive to non-Gaussian noise, so few tracking errors can spoil the result; (2) it requires all trajectories to have the same length, so partial occlusion and disocclusion can actually not be handled. Recent works suggest ways to deal with these problems [19,20], but as the problems are inherent to factorization, this can only be successful up to a certain degree. For instance, it is still required that a sufficiently large subset of trajectories exists for the whole time line of the shot.

There are a few works which analyze point trajectories outside the factorization setting [7,21,22,23]. Like the proposed method, these techniques do not require a dominant subset of trajectories covering the full time line, and apart from [21], which analyzes trajectories but runs the clustering on a single-frame basis, these methods provide temporally consistent clusters. Technically, however, they are very different from our approach, with regard to the density of trajectories, how the distance between trajectories is defined, and in the algorithm used for clustering.

Trajectory clustering is not restricted to the domain of object segmentation. For instance, it has been used for learning traffic models in [24].

### 3 Point Tracking and Affinities between Trajectories

We obtain point trajectories by running the optical flow based tracker in [2] on a sequence. Fig. 3 demonstrates the most important properties of this tracker. Clearly, the coverage of the image by tracked points is much denser than with usual keypoint trackers. This is very advantageous for our task, as this allows us to assign labels far more densely than in previous approaches. Moreover, the denser coverage of trajectories will enable local reasoning from motion similarities as well as the introduction of spatial regularity constraints in the clustering method.



**Fig. 3. From left to right:** Initial points in the first frame and tracked points in frame 211 and 400. Color indicates the age of the tracks. The scale goes from blue (young) over green, yellow, and red to magenta (oldest). The red points on the right person have been tracked since the person appeared behind the wall. The figure is best viewed in color.

Fig. 3 also reveals that points can be tracked over very long time intervals. A few points on the wall were tracked for all the 400 frames. The other tracks are younger because almost all points in this scene have become occluded. The person on the right appeared behind the wall and was initially quite far away from the camera. The initial points from that time have been tracked to the last frame and are visible as red spots among all the other tracks that were initialized later due to the scaling of the person.

Clearly, trajectories are asynchronous, i.e., they cover different temporal windows in a shot. This is especially true if the shot contains fast motion and large occluded areas. If we only selected the set of trajectories that survived the whole shot, this set would be very small or even empty and we would miss many dominant objects in the scene. So rather than picking a fully compatible subset, we define pairwise affinities between all trajectories that share at least one frame. The affinities define a graph upon which we run spectral clustering. Due to transitivity, even tracks that do not share common frames can be linked up as long as there is a path in the graph that connects them.

According to the Gestalt principle of common fate, we should assign high affinities to pairs of points that move together. However, two persons walking next to each other share the same motion although they are different objects. We have to take into account that there are situations where we cannot tell two objects apart. The actual information is not in the common motion but in motion differences. As soon as one of the persons moves in another direction from the other one, we get a very clear signal that these two areas in the image do not belong together.

We define distances and affinities such that they best exploit this information. Regarding two trajectories  $A$  and  $B$ , we consider the instant, where the motion of the two points is most dissimilar:

$$d^2(A, B) = \max_t d_t^2(A, B). \quad (1)$$

Pairwise distances can only compare the compatibility of trajectories on the basis of translational motion models. To estimate the parameters of a more

general motion model, we would have to consider triplets or even larger groups of points, which is intractable. Another way is to estimate such models beforehand using a RANSAC procedure to deal with the fact that we do not know yet which points share the same motion model [7]. However, especially in case of many smaller regions, one needs many samples to ensure a set of points without outliers with high probability. Instead, we rely here on the fact that translational models are a good approximation for spatially close points and introduce a proper normalization in order to reduce the negative effects of this approximation.

We define the distance between two trajectories at a particular instant  $t$  as:

$$d_t^2(A, B) = d_{\text{sp}}(A, B) \frac{(u_t^A - u_t^B)^2 + (v_t^A - v_t^B)^2}{5\sigma_t^2}. \quad (2)$$

$d_{\text{sp}}(A, B)$  denotes the average spatial Euclidean distance of  $A$  and  $B$  in the common time window. Multiplying with the spatial distance ensures that only proximate points can generate high affinities. Note that due to transitivity, points that are far apart can still be assigned to the same cluster even though their pairwise affinity is small.  $u_t := x_{t+5} - x_t$  and  $v_t := y_{t+5} - y_t$  denote the motion of a point aggregated over 5 frames. This averaging adds some further accuracy to the motion estimates. If less than 5 frames are covered we average over the frames that are available. Another important detail is the normalization of the distance by

$$\sigma_t = \min_{a \in \{A, B\}} \sum_{t'=1}^5 \sigma(x_{t+t'}^a, y_{t+t'}^a, t + t'), \quad (3)$$

where  $\sigma : \mathbb{R}^3 \rightarrow \mathbb{R}$  denotes the local flow variation field. It can be considered a local version of the optical flow variance in each frame and is computed with linear diffusion where smoothing is reduced at discontinuities in the optical flow.

The normalization by  $\sigma_t$  is very important to deal with both fast and slow motion. If there is hardly any motion in a scene, a motion difference of 2 pixels is a lot, whereas the same motion difference is negligible in a scene with fast motion. As scaling and rotation will cause small motion differences even locally, it is important to consider these differences in the context of the overall motion. Considering the local rather than the global variance of the optical flow makes a difference if at least three motion clusters appear in the scene. The motion difference between two of them could be small, while the other differences are large.

We use the standard exponential and a fixed scale  $\lambda = 0.1$  to turn the distances  $d^2(A, B)$  into affinities

$$w(A, B) = \exp(-\lambda d^2(A, B)) \quad (4)$$

yielding an  $n \times n$  affinity matrix  $W$  for the whole shot, where  $n$  is the total number of trajectories.



**Fig. 4. From left to right, top to bottom: (a)** Input frame. **(b-h)** The first 7 of  $m = 13$  eigenvectors. Clearly, the eigenvectors are not piecewise constant but show smooth transitions within the object regions. However, discontinuities in the eigenvectors correspond to object boundaries very well. This information needs to be exploited in the final clustering procedure.

## 4 Spectral Clustering with Spatial Regularity

Given an affinity matrix, the most common clustering techniques are agglomerative clustering, which runs a greedy strategy, and spectral clustering, which maps the points into a feature space where more traditional clustering algorithms like k-means can be employed. While the mapping in spectral clustering is a globally optimal step, the successive step that yields the final clusters is like all general clustering susceptible to local minima. We rely on the eigendecomposition of the normalized graph Laplacian to obtain the mapping and elaborate on deriving good clusters from the resulting eigenvectors. The setting we propose also includes model selection, i.e., it decides on the optimum number of clusters.

Let  $D$  be an  $n \times n$  diagonal matrix with entries  $d_a = \sum_b w(a, b)$ . The Laplacian eigenmap is obtained by an eigendecomposition of the normalized Laplacian

$$V^T AV = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}} \quad (5)$$

and keeping the eigenvectors  $\mathbf{v}_0, \dots, \mathbf{v}_m$  corresponding to the  $m + 1$  smallest eigenvalues  $\lambda_0, \dots, \lambda_m$ . As  $\lambda_0 = 0$  and  $\mathbf{v}_0$  is a constant vector, this pair can be ignored. We choose  $m$  such that we keep all  $\lambda < 0.2$ . The exact choice of this threshold is not critical as long as it is not too low, since the actual model selection is done in eigenvector space. Since  $m \ll n$ , the eigendecomposition can be efficiently computed using the Lanczos method. We normalize all eigenvectors  $\mathbf{v}_i$  to a range between 0 and 1.

In case of ideal data (distinct translational motion, no tracking errors), the mapping yields  $m = k - 1$  piecewise constant eigenvectors and the  $k$  clusters can be extracted by simple thresholding [5]. However, the eigenvectors are usually not that clean, as shown in Fig. 4. The eigenvectors typically show smooth transitions within a region and more or less clear edges between regions. Standard k-means

cannot properly deal with this setting either, since smooth transitions get approximated by multiple constant functions, thus leading to an over-segmentation. At the same time the optimum number of clusters  $K$  is by no means obvious as clusters are represented by many eigenvectors.

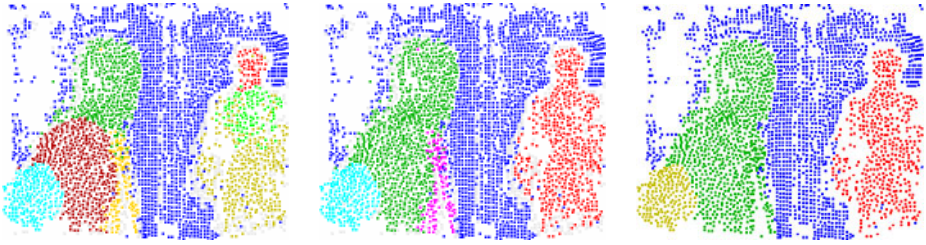
As a remedy to both problems, we suggest minimizing an energy function that comprises a spatial regularity term. Let  $v_i^a$  denote the  $a$ th component of the  $i$ th eigenvector and  $\mathbf{v}^a$  the vector composed of the  $a$ th components of all  $m$  eigenvectors. Index  $a$  corresponds to a distinct trajectory. Let  $\mathcal{N}(a)$  be a set of neighboring trajectories based on the average spatial distance of trajectories. We seek to choose the total number of clusters  $K$  and the assignments  $\pi^a \in \{1, \dots, K\}$  such that the following energy is minimized:

$$E(\pi, K) = \sum_a \sum_{k=1}^K \delta_{\pi^a, k} \|\mathbf{v}^a - \mu_k\|_\lambda^2 + \nu \sum_a \sum_{b \in \mathcal{N}(a)} \frac{1 - \delta_{\pi^a, \pi^b}}{|\mathbf{v}^a - \mathbf{v}^b|_2^2} \quad (6)$$

The first term is the unary cost that is minimized by k-means, where  $\mu_k$  denotes the centroid of cluster  $k$ . The norm  $\|\cdot\|_\lambda$  is defined as  $\|\mathbf{v}^a - \mu\|_\lambda = \sum_i (v_i^a - \mu_i)^2 / \lambda_i$ , i.e., each eigenvector is weighted by the inverse of the square root of its corresponding eigenvalue. This weighting is common in spectral clustering as eigenvectors that separate more distinct clusters correspond to smaller eigenvalues [25].

Clearly, if we do not restrict  $K$  or add a penalty for additional clusters, each trajectory will be assigned its own cluster and we will get a severe over-segmentation. The second term in (6) serves as a regularizer penalizing the spatial boundaries between clusters. The penalty is weighted by the inverse differences of the eigenvectors along these boundaries. If there are clear discontinuities along the boundary of two clusters, the penalty for this boundary will be very small. In contrast, boundaries within a smooth area are penalized far more heavily, which avoids splitting clusters at arbitrary locations due to smooth transitions in the eigenvectors. The parameter  $\nu$  steers the tradeoff between the two terms. We obtain good results in various scenes by fixing  $\nu = \frac{1}{2}$ .

Minimizing (6) is problematic due to many local minima. We propose a heuristic that avoids such local minima. For a fixed  $K$ , we first run k-means with 10 random initializations. Additionally, we generate proposals by running hierarchical 2-means clustering and selecting the 20 best solutions from the tree. We run k-means on these 20 proposals and select the best among all 30 proposals. Up to this point we consider only the first term in (6), since the proposals are generated only according to this criterion. The idea is that for a large enough  $K$  we will get an over-segmentation that comprises roughly the boundaries of the true clusters. In a next step we consider merging moves. We successively consider the pair of clusters that when merged leads to the largest reduction of (6) including the second term. Merging is stopped if the energy cannot be further minimized. Finally, we run gradient descent to locally optimize the assignments. This last step mainly refines the assignments along boundaries. The whole procedure is run for all  $K \in \{1, \dots, 2m\}$  and we pick the solution with the smallest energy.



**Fig. 5.** **Left: (a)** Best  $k$ -means proposal obtained for  $K = 9$ . Over-segmentation due to smooth transitions in eigenvectors. **Center: (b)** Remaining 5 clusters after choosing the best merging proposals. **Right: (c)** Final segmentation after merging using affine motion models. Another cluster boundary that was due to the fast 3D rotation of the left person has been removed. The only remaining clusters are the background, the two persons, and the articulated arm of the left person.

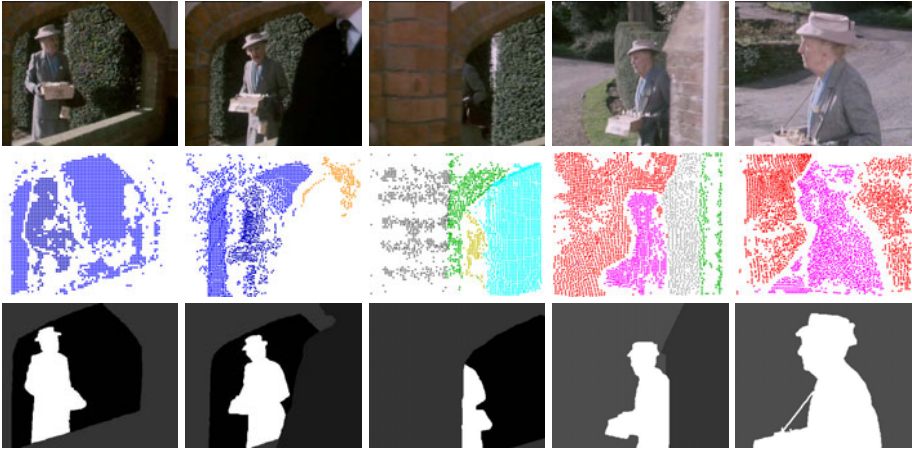
Finally, we run a postprocessing step that merges clusters according to the mutual fit of their affine motion models. This postprocessing step is not absolutely necessary, but corrects a few over-segmentation errors. Fig. 5 shows the clusters obtained by  $k$ -means, after merging clusters of the  $k$ -means proposal, and after the postprocessing step.

## 5 Experimental Evaluation

### 5.1 Dataset and Evaluation Method

While qualitative examples often reveal more details of a method than pure numbers, scientific research always benefits from exact measurement. The task of motion segmentation currently lacks a compelling benchmark dataset to produce such measurements and to compare among methods. While the Hopkins 155 dataset [8] has clearly boosted research in multi-body factorization, it is much too specialized for these types of methods, and particularly the checkerboard sequences do not correspond to natural scenes. To this end, we have annotated 26 sequences, among them shots from detective stories and the 10 car and 2 people sequences from Hopkins 155, with a total of 189 annotated frames. The annotation is dense in space and sparse in time, with more frames being annotated at the beginning of a shot to allow also for the evaluation of methods that do not work well with long sequences. There are four evaluation modes. The first three expect the methods to be run only on the first 10, 50, and 200 frames, whereas for the last all available frames should be considered. It is planned to successively extend the dataset by more sequences to avoid over-fitting issues in the long run. An example of the annotation is shown in Fig. 6. This dataset is publicly available.

The evaluation tool yields 5 numbers for each sequence, which are then averaged across all sequences. The first number is the **density** of the points for which a cluster label is reported. Higher densities indicate more information extracted



**Fig. 6.** Frames 1, 110, 135, 170, 250 of a shot from *Miss Marple: Murder at the vicarage* together with our clusters and the ground truth regions. There is much occlusion in this sequence as Miss Marple is occluded by the passing inspector and then by the building. Our approach can link tracks of partially occluded but not of totally occluded objects. A linkage of these clusters is likely to be possible based on the appearance of the clusters and possibly some dynamic model.

from the sequences and increase the risk of misclassifying pixels. The **overall clustering error** is the number of bad labels over the total number of labels on a per-pixel basis. The tool optimally assigns clusters to ground truth regions. Multiple clusters can be assigned to the same region to avoid high penalties for over-segmentations that actually make sense. For instance, the head of a person could be detected as a separate cluster even though only the whole person is annotated in the dataset. All points covering their assigned region are counted as good labels, all others count as bad labels. In some sequences, objects are marked that are easy to confuse due to their size or very little motion information. A penalty matrix defined for each sequence assigns smaller penalty to such confusions. The **average clustering error** is similar to the overall error but averages across regions after computing the error for each region separately. Usually the average error is much higher than the overall error, since smaller objects are more difficult to detect, confused regions always pay the full penalty, and not covering an object yields a 100% error for that region.

Since the above evaluation method allows for cheating by producing a severe over-segmentation, we also report the **over-segmentation error**, i.e., the number of clusters merged to fit the ground truth regions. Methods reporting good numbers with a very high over-segmentation error should be considered with care.

As the actual motivation for motion segmentation is the unsupervised extraction of objects, we finally report the number of regions covered with less than 10% error. One region is subtracted per sequence to account for the background.



## 5.2 Results

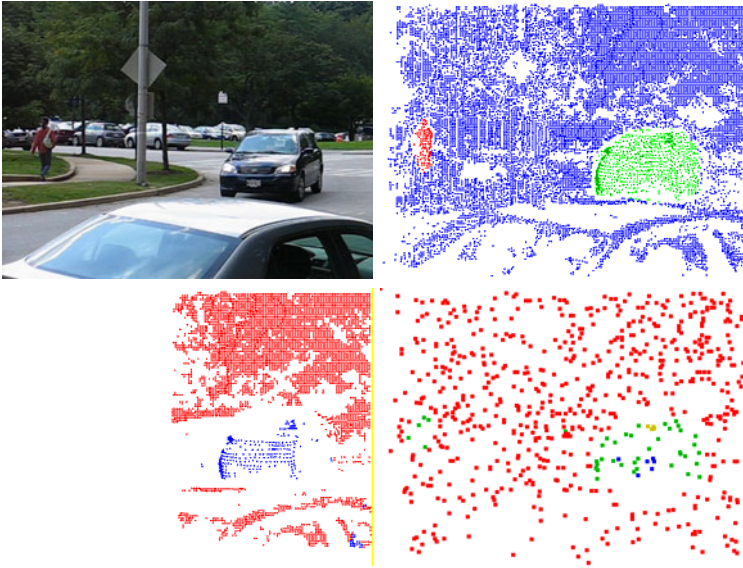
Apart from the numbers of the proposed technique we also report numbers for Generalized PCA (GPCA), Local Subspace Affinity (LSA) [18], and RANSAC using the code provided with the Hopkins dataset [8]. We also show results for the factorization method in [19], which can deal with either incomplete or corrupted trajectories (ALC). When running these methods, we use the same trajectories as for our own method. Except for ALC with incomplete tracks, all these techniques require the tracks to have full length, so we restrict the available set of tracks accordingly. For this reason, the density with these methods is considerably smaller, especially when more frames are taken into account and the areas of occlusion and disocclusion grow bigger. Moreover, all these methods ask for the number of regions to be given in advance. We give them the correct number, whereas we select the model in our approach automatically. Since ALC gets intractably slow when considering more than 1000 trajectories (see Table 1), we randomly subsampled the tracks for this method by a factor 16. In Table 2 we multiply the density again by this factor to make the results comparable.

**Table 1.** Computation times for the people1 sequence of the Hopkins dataset considering only the first 10 frames. When running ALC we randomly subsampled the tracks by a factor 16 to have more tractable computation times.

	tracks	time
our method	15486	497s
GPCA	12060	2963s
LSA	12060	38614s
RANSAC	12060	15s
ALC	957	22837s

**Table 2.** Evaluation results. The sequence marple7 was ignored in the entry marked with \* as the computation took more than 800 hours.

	Density	overall error	average error	over-segmentation	extracted objects
First 10 frames (26 sequences)					
our method	3.34%	7.75%	25.01%	0.54	24
GPCA	2.98%	14.28%	29.44%	0.65	12
LSA	2.98%	19.69%	39.76%	0.92	6
RANSAC	2.98%	13.39%	26.11%	0.50	15
ALC corrupted	2.98%	7.88%	24.05%	0.15	26
ALC incomplete	3.34%	11.20%	26.73%	0.54	19
First 50 frames (15 sequences)					
our method	3.27%	7.13%	34.76%	0.53	9
ALC corrupted	1.53%	7.91%	42.13%	0.36	8
ALC incomplete	3.27%	16.42%	49.05%	6.07	2
First 200 frames (7 sequences)					
our method	3.43%	7.64%	31.14%	3.14	7
ALC corrupted	0.20%	0.00%	74.52%	0.40	1
ALC incomplete	3.43%	19.33%	50.98%	54.57	0
All available frames (26 sequences)					
our method	3.31%	6.82%	27.34%	1.77	27
ALC corrupted	0.99%	5.32%	52.76%	0.10	15
ALC incomplete*	3.29%	14.93%	43.14%	18.80	5



**Fig. 7. From left to right:** (a) Frame 40 of the cars4 sequence from the Hopkins dataset. (b) The proposed method densely covers the image and extracts both the car and the person correctly. (c) RANSAC (like all traditional factorization methods) can assign labels only to complete tracks. Thus large parts of the image are not covered. (d) ALC with incomplete trajectories [19] densely covers the image, but has problems assigning the right labels.

Clearly, the more traditional methods like GPCA, LSA, and RANSAC do not perform well on this dataset (which comprises a considerable number of sequences from the Hopkins dataset). Even when considering only 10 frames, i.e. there is only little occlusion, the error is much larger than for the proposed approach. The 10-frame result for ALC with a correction for corrupted tracks is quite good and comparable to ours with some advantages with regard to over-segmentation and extracted objects. This is mainly due to the correct number of regions given to ALC.

As the number of frames is increased, the density of ALC decreases and its performance goes down. With more occlusions, ALC with incomplete tracks becomes interesting, as it is the only method in this comparison apart from ours that can exploit all trajectories. However, its ability to handle sparse trajectories is limited. ALC still needs a sufficiently large set of complete tracks in order to extrapolate the missing entries, whereas the approach described in this paper just requires some overlapping pieces of trajectories to cluster them together. We see a larger over-segmentation error for the longer sequences, as occlusion introduces ambiguities and makes the clustering problem generally harder, but at the same time we obtain more information about the tracked objects. Moreover, by considering more frames, objects that were static in the first frames can be

extracted due to their motion in a later frame. We obtain the smallest overall error and can extract the most objects when considering all the frames in a shot.

Fig. 7 highlights the qualitative differences between the types of methods. The proposed method can densely cover the full image with cluster labels despite significant occlusions, and errors in the trajectories are handled well. Recent factorization methods like ALC with correction for corrupted tracks work quite well for the subset of complete tracks, but they cannot produce labels for points that are not visible in all frames. ALC for incomplete tracks can generally cover the whole image with labels, but as this is achieved by extrapolating missing entries, lots of errors occur. In case ALC cannot find the given number of regions, it uses an MDL criterion, which leads to a very high over-segmentation error.

The density of our approach is still far from 100%. This is mainly due to efficiency considerations, as the tracker in [2] could also produce denser trajectories. However, the trajectories already cover the image domain without too many larger gaps. In this paper, we did without static cues to keep the paper uncluttered. Given these point labels, however, it actually should be quite straightforward to obtain a dense segmentation by considering color or boundary information.

## 6 Conclusions

We have presented a technique for object-level segmentation in a pure bottom-up fashion by exploiting long term motion cues. Motion information is aggregated over the whole shot to assign labels also to objects that are static in a large part of the sequence. Occlusion and disocclusion is naturally handled by this approach, which allows to gather information about an object from multiple aspects. This kind of motion segmentation is far more general than most previous techniques based on two-frame optical flow or a sparse subset of complete trajectories. We believe that such a general setting is very relevant, as it will ultimately enable unsupervised learning of objects from appropriate video data. We hope that by providing a benchmark dataset that comprises a variety of easier and harder sequences, we can foster progress in this field.

## References

1. Spelke, E.: Principles of object perception. *Cognitive Science* 14, 29–56 (1990)
2. Sundaram, N., Brox, T., Keutzer, K.: Dense point trajectories by GPU-accelerated large displacement optical flow. In: *European Conf. on Computer Vision*. LNCS, Springer, Heidelberg (2010)
3. Brox, T., Malik, J.: Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (to appear)
4. Sand, P., Teller, S.: Particle video: long-range motion estimation using point trajectories. *International Journal of Computer Vision* 80, 72–91 (2008)
5. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 888–905 (2000)

6. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: *Advances in Neural Information Processing Systems* (2002)
7. Sivic, J., Schaffalitzky, F., Zisserman, A.: Object level grouping for video shots. *International Journal of Computer Vision* 67, 189–210 (2006)
8. Tron, R., Vidal, R.: A benchmark for the comparison of 3-D motion segmentation algorithms. In: *Int. Conf. on Computer Vision and Pattern Recognition* (2007)
9. Koffka, K.: *Principles of Gestalt Psychology*. Hartcourt Brace Jovanovich, New York (1935)
10. Wang, J.Y.A., Adelson, E.H.: Representing moving images with layers. *IEEE Transactions on Image Processing* 3, 625–638 (1994)
11. Weiss, Y.: Smoothness in layers: motion segmentation using nonparametric mixture estimation. In: *Int. Conf. on Computer Vision and Pattern Recognition*, pp. 520–527 (1997)
12. Shi, J., Malik, J.: Motion segmentation and tracking using normalized cuts. In: *Proc. 6th International Conference on Computer Vision, Bombay, India*, pp. 1154–1160 (1998)
13. Cremers, D., Soatto, S.: Motion competition: A variational framework for piecewise parametric motion segmentation. *International Journal of Computer Vision* 62, 249–265 (2005)
14. Xiao, J., Shah, M.: Motion layer extraction in the presence of occlusion using graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 1644–1659 (2005)
15. Pawan Kumar, M., Torr, P., Zisserman, A.: Learning layered motion segmentations of video. *International Journal of Computer Vision* 76, 301–319 (2008)
16. Smith, P., Drummond, T., Cipolla, R.: Layered motion segmentation and depth ordering by tracking edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 479–494 (2004)
17. Costeira, J., Kanade, T.: A multi-body factorization method for motion analysis. In: *Int. Conf. on Computer Vision*, pp. 1071–1076 (1995)
18. Yan, J., Pollefeys, M.: A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3954, pp. 94–106. Springer, Heidelberg (2006)
19. Rao, S.R., Tron, R., Vidal, R., Ma, Y.: Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In: *Int. Conf. on Computer Vision and Pattern Recognition* (2008)
20. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: *Int. Conf. on Computer Vision and Pattern Recognition* (2009)
21. Brostow, G., Cipolla, R.: Unsupervised Bayesian detection of independent motion in crowds. In: *Int. Conf. on Computer Vision and Pattern Recognition* (2006)
22. Cheriyyadat, A., Radke, R.: Non-negative matrix factorization of partial track data for motion segmentation. In: *Int. Conf. on Computer Vision* (2009)
23. Fradet, M., Robert, P., Pérez, P.: Clustering point trajectories with various lifespans. In: *Proc. European Conference on Visual Media Production* (2009)
24. Wang, X., Tieu, K., Grimson, E.: Learning semantic scene models by trajectory analysis. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3953, pp. 110–123. Springer, Heidelberg (2006)
25. Belongie, S., Malik, J.: Finding boundaries in natural images: A new method using point descriptors and area completion. In: Burkhardt, H.-J., Neumann, B. (eds.) *ECCV 1998*. LNCS, vol. 1406, pp. 751–766. Springer, Heidelberg (1998)

# Exploiting Repetitive Object Patterns for Model Compression and Completion

Luciano Spinello<sup>1,2</sup>, Rudolph Triebel<sup>2</sup>, Dizan Vasquez<sup>3</sup>,  
Kai O. Arras<sup>1</sup>, and Roland Siegwart<sup>2</sup>

<sup>1</sup> Social Robotics Lab, University of Freiburg, Germany

<sup>2</sup> Autonomous Systems Lab, ETH Zurich, Switzerland

<sup>3</sup> ITESM Campus Cuernavaca, Mexico

**Abstract.** Many man-made and natural structures consist of similar elements arranged in regular patterns. In this paper we present an unsupervised approach for discovering and reasoning on repetitive patterns of objects in a single image. We propose an unsupervised detection technique based on a voting scheme of image descriptors. We then introduce the concept of *latticelets*: minimal sets of arcs that generalize the connectivity of repetitive patterns. Latticelets are used for building polygonal cycles where the smallest cycles define the sought groups of repetitive elements. The proposed method can be used for pattern prediction and completion and high-level image compression. Conditional Random Fields are used as a formalism to predict the location of elements at places where they are partially occluded or detected with very low confidence. Model compression is achieved by extracting and efficiently representing the repetitive structures in the image. Our method has been tested on simulated and real data and the quantitative and qualitative result show the effectiveness of the approach.

## 1 Introduction

Man-made and natural environments frequently contain sets of similar basic elements that are arranged in regular patterns. Examples include architectural elements such as windows, pillars, arcs, or structures in urban environments such as equidistant trees, street lights, or similar houses built in a regular distance to each other. There are at least two applications where models of repetitive structures are useful pieces of information: occlusion handling and data compression. For the former, pattern information can be used to predict the shape and position of occluded or low confidence detections of objects in the same scene. This introduces a scheme in which low-level detections are mutually reinforced by high-level model information. For model compression, representing the repetitive structure by a generalized object and pattern description makes it possible to represent the structure of interest in the image very efficiently.

In this paper, we present a technique to find such repetitive patterns in an unsupervised fashion and to exploit this information for occlusion handling and compression. Specifically, we evaluate our method on the problem of building facade analysis.

The contributions of this paper are:

1. Unsupervised detection of mutually similar objects. Closed contours are extracted and robustly matched using a growing codebook approach inspired by the Implicit Shape Models (ISM) [1].

2. Analysis of pattern repetitions by the concept of *latticelets*: a selected set of frequent distances between elements of the same object category in the Cartesian plane. Latticelets are generalizations of the repetition pattern.
3. A probabilistic method to geometrically analyze cyclic element repetitions. Using Conditional Random Fields (CRF) [2], the method infers missing object occurrences in case of weak hypotheses. Element detection probability and geometrical neighborhood consistency are used as node and edge features.

Our method is a general procedure to discover and reason on repetitive patterns, not restricted to images. The only requirement is that a method for detecting similar objects in a scene is available and that a suitable latticelet parameterization is available in the space of interest, e.g. the image or Cartesian space.

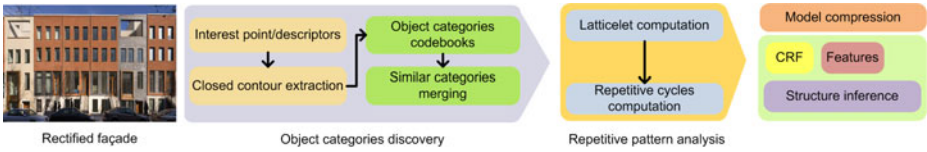
To the authors' best knowledge, there is no other work in the literature that pursues the same goals addressing the problem in a principled way.

This paper is organized as follows: the next section discusses related work. Section 3 gives an overview of our technique while in Section 4, the process of element discovery is explained. Section 5 presents the way we analyze repetitive patterns and Section 6 describes how to use CRFs for the task of repetitive structure inference. Section 7 shows how to obtain an high-level image compression with the proposed method. In Section 8 the quantitative and qualitative experiments are presented followed by the conclusions in Section 9.

## 2 Related Work

In this work we specifically analyze repetitions from a single static image. The work of [3] uses Bayesian reasoning to model buildings by architectural primitives such as windows or doors parametrized by priors and assembled together like a 'Lego kit'. The work of [4] interprets facades by detecting windows with an ISM approach. A predefined training set is provided. Both works address the problem with a Markov Chain Monte Carlo (MCMC) technique. Unlike our approach, they do not exploit information on the connectivity between the detected elements. Our work uses ISM in an unsupervised fashion without a priori knowledge. We consider closed contours to create codebooks that generalize the appearance of repeated elements. Thereby, we are able to recognize such elements with high appearance variability thanks to the Hough-voting scheme. In the field of computer graphics, grammar based procedural modeling [5,6,7] has been formally introduced to describe a way of representing man-made buildings. Most of these works do not discover patterns but reconstruct the 3D appearance of the facade and require human intervention.

Approaches based on RANSAC [8] and the Hough transform [9] have been used to find regular, planar patterns. More sophisticated methods relax the assumption of the regular pattern using Near-Regular Textures (NRT) [10,11]. Similar to our work is [12] in which the authors propose a method to find repeated patterns in a facade by using NRT with MCMC optimization using rules of intersection between elements. They are able to extract a single pattern based on a 4-connectivity lattice. Our approach allows detection of arbitrary patterns without relying on a fixed model. Further, it can detect multiple object categories and associate for each category multiple repetition patterns.



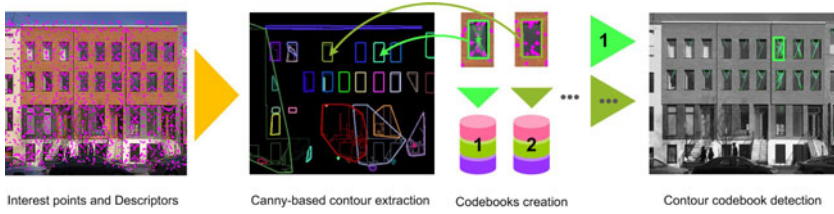
**Fig. 1.** Schematic overview of the algorithm

### 3 Overview

The first step of our algorithm (see Fig. 1) is to compute a set of standard descriptors on a given input image. Then, we compute closed contours that represent the candidates for repetitive objects such as windows or pillars. The key idea is that we do not classify these objects using a model that was previously learned from training data, but instead, obtain evidence of their occurrence by extracting similarities directly from the given scene. The advantage of this is twofold: first, we are independent of a previously hand-labeled training data set. Second, by grouping similar objects into categories and considering only those categories with at least two object instances, we can filter out outlier categories for which no repetitive pattern can be found. Our measure of mutual similarity is based on the object detection approach by Leibe *et al.* [11]. In the next step, we analyze repetitive patterns inside each category. This is done by analyzing the Euclidean distances between elements in the image accumulated in a frequency map. These relative positions are represented as edges in a lattice graph in which nodes represent objects positions. The most dominant edges by which all nodes in this graph can be connected are found using a Minimum Spanning Tree algorithm and grouped into a set that we call latticelet. For reasoning on higher-level repetitions we extract a set of polygonal repetitions composed of latticelet arcs. Such polygonal repetitions are used to build a graph for predicting the position of occluded or weakly detected elements. An inference engine based on CRFs is used to determine if the occurrence of an object instance at a predicted position is likely or not. In an image compression application, we use a visual template of each object category, the medium background color and the lattice structure to efficiently store and retrieve a given input image.

### 4 Extraction of Mutually Similar Object Instances

In this section we explain the process of discovering repetitive elements present in an image based on closed contours. As first step of the algorithm, Shape Context descriptors [13] are computed at Hessian-Laplace interest points. Contours are computed by using the binary output of the Canny edge detector [14] encoded via Freeman chain code [15]. We refer to the content in each contour as an object instance  $O_e$ . Matching contours in real world images can be very hard due to shadows and low contrast areas. We therefore employ an Implicit Shape Model-like (ISM) technique in which the contours act as containers to define a codebook of included descriptors. This way, we can robustly match objects. In summary, an ISM consists of a set of local region descriptors, called *codebook*, and a set of displacements, usually named *votes*, for each descriptor.



**Fig. 2.** Extraction of mutually similar objects. For each closed contour, a codebook of descriptors is created that contains relative displacements to the object centers (votes). Then, the descriptors of each object are matched against the descriptors in the image.

The idea is that each descriptor can be found at different positions inside an object and at different scales. Thus, a vote points from the position of a matched descriptor to the center of the object as it was associated in the codebook construction. In our case all the descriptors found inside a contour are included in the codebook  $\mathcal{C}_e$  as well as the relative displacement of the respective interest points with respect to the center of the contour. To retrieve objects repetitions we match objects in the following way:

1. All descriptors found in the image are matched against an object's codebook  $\mathcal{C}_e$ . Those with a Euclidean distance to the best match in  $\mathcal{C}_e$  that is bigger than a threshold  $\theta_d$  are discarded.
2. Votes casted by the matching descriptors are collected in a 2D voting space
3. We use mean shift mode estimation to find the object center from all votes. This is referred to as an object hypothesis.

To select valid hypotheses we propose a quality function that balances the strength of the votes with their spatial origin. Votes are accumulated in a circular histogram around the hypothetical object center. The detection quality function is given by:

$$q_i = w_a \cdot \frac{f_h(\alpha_i, \alpha_e)}{f_h(\alpha_e, \alpha_e)} + (1 - w_a) \cdot \frac{s_i}{s_e} \quad q_i \in [0, 1] \quad (1)$$

where  $\alpha_e$  is the vote orientation histogram of the object  $\mathcal{C}_e$ ;  $\alpha_i$  is the vote orientation histogram of the hypothesis  $i$ ;  $f_h$  is a function that applies an *AND* operator between the bins of two histograms and sums the resulting not empty bins.  $s_i, s_e$  are respectively the score (number of votes received for the hypothesis) and the score of  $O_e$ .  $w_a$  is the bias that is introduced between the two members. This is a simplified version of the cost function explained in [16]. Detected objects are selected by a simple minimum threshold  $\theta_q$  on the detection quality  $q_i$ . All the objects matching with  $O_e$  constitute the object category  $\tau$  that is defined by a codebook composed by descriptors that contributed to each match and all the entries of  $\mathcal{C}_e$ . Thus, a more complete description of the visual variability of the initial object instance  $O_e$  is achieved. It is important to notice that it is not required that every object in the image has a closed contour as soon as there is at least one of its category. In other words: if an image of a facade contains several windows of the same type, only one of them is required to have a closed contour. In this work we aim to match objects with the same scale. Same objects present at different scales in the image are treated as different object categories.

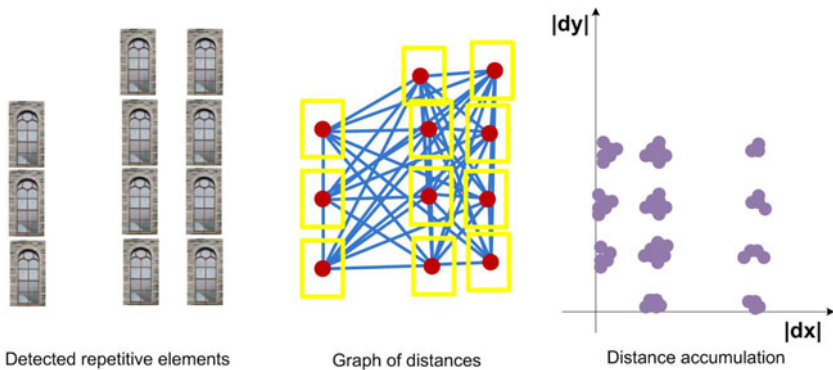


As a last step we use an hierarchical agglomerative clustering with average linkage to group visually similar categories by using a measure described by their codebook entries  $d(\tau_{\mathcal{C}}^i, \tau_{\mathcal{C}}^j) = \frac{L(\tau_{\mathcal{C}}^i, \tau_{\mathcal{C}}^j)}{\min(|\tau_{\mathcal{C}}^i|, |\tau_{\mathcal{C}}^j|)}$  where  $L$  computes the number of corresponding descriptors from the two codebooks with a Euclidean distance of less than  $\theta_d$  and  $|\tau_{\mathcal{C}}^i|$  the number of codebook entries.

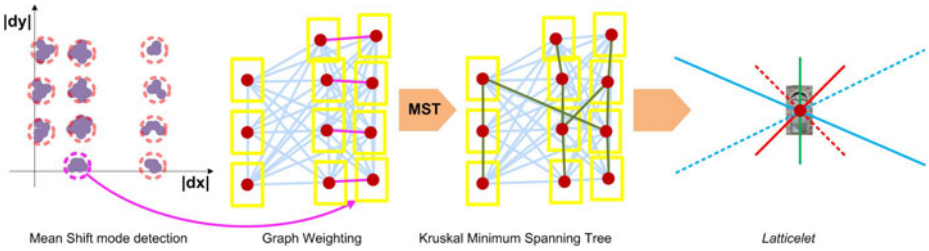
## 5 Analysis of Repetitive Objects

### 5.1 Latticelets

In this section we introduce the space frequency analysis for the discovered object categories. We name the detected object locations in the image as *nodes*. In order to analyze the repetition pattern of each object category we build a complete graph that connects all the nodes. Our aim is to select in this graph edges that have a repeated length and orientation. Moreover, we require our arc selection to include all the nodes. Our proposed solution is based on the use of a Minimum Spanning Tree (MST). From the complete graph we build a frequency map (see scheme Fig. 3 and Fig. 4), in which we store the distances  $|dx|, |dy|$  in pixels between nodes of the graph. The map represents the complete distance distribution between the nodes. We therefore have to select from this map the most representative modes. In order to estimate local density maxima in the frequency map we employ a two dimensional mean shift algorithm, with a simple circular kernel. Each convergence mode is expressed by a point in the map  $d\hat{x}, d\hat{y}$  and its score repetitiveness that is given by the number of points contributing to the basin of attraction. All the graph edges that contribute to each mode convergence are then labeled with their associated distance. At the end of this process we have obtained a graph in which the distances between the nodes have been relaxed by averaging similar consistent distances/orientations. Each edge is tagged with its repetitiveness score.



**Fig. 3.** Latticelet discovery process. Objects of the same category are detected. A complete graph is built and the relative distances are accumulated in the Cartesian plane.



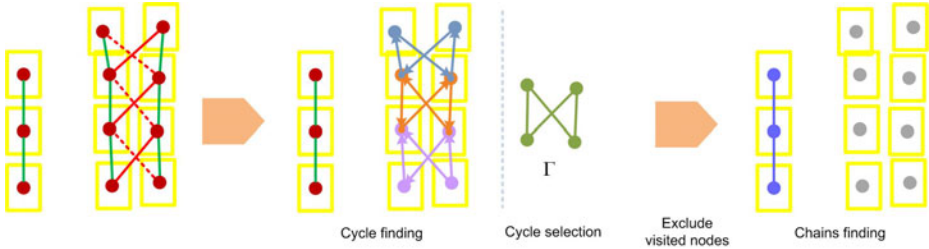
**Fig. 4.** Repetitive distances in  $x$  and  $y$  are clustered via mean-shift, the arcs are reweighted by their mode convergency score. The solid and dotted lines in the latticelet figure represent the possible directions expressed by the selected  $|dx|$  and  $|dy|$ .

As last step of this processing we employ Kruskal’s algorithm [17] to find the minimum spanning tree by using the nodes, their edge connectivity and the weight of the arcs. The resulting tree represents the most repetitive arcs sufficient to connect all the nodes. In order to compact the information we select each kind of arc just once. We call it latticelet, the minimal set of repetitive arcs that are needed to represent the original lattice. Each object category is associated to a latticelet that generalize its repetition pattern. Our method is able to cope with small perspective distortions thanks to the relaxation step. For larger deviations from a fronto-parallel image view, the problem of perspective estimation can be naturally decoupled from the one of analyzing repetitive patterns. The problem of image rectification could be addressed with many existing methods (e.g. [18]) that are far beyond the scope of this paper.

## 5.2 Cycles and Chains

Latticelets contain very local information, they explain the direction of a possible predicted element from a given position. In order to incorporate higher level knowledge of the repetitive pattern of the neighborhood, we use cycles composed of latticelets arcs. Our aim is to find minimal size repetitive polygons. They provide the effective object repetition that is used in later stages to obtain prediction and simplification. For each category we sort the the weight of its latticelet arcs and we select the one with highest weight. We compose a new graph by using the selected arc to build connection between nodes and compute the smallest available cycle by computing its girth (i.e. length)  $\gamma$ .

A cycle  $\Gamma$  is computed by using an approach based on a Breadth-first Search algorithm. Starting from a node of choice in the graph, arcs are followed once, and nodes are marked with their number of visits. A cycle is found as soon as the number of visits for a node reaches two. This is done for all the nodes present in the object category detection set. We then collect all the cycles, and we select the one with the smallest number of nodes. We create a graph by using the connectivity offered by  $\Gamma$  and mark as removed the nodes that are connected by it. Thus, we add another latticelet arc until all the nodes are connected or all the latticelet arcs are used. We obtained a polygon set composed of frequent displacements suitable to describe the object distribution in the image (see scheme Fig. 5) and to generalize higher orders repetitions. An object category is therefore associated to  $k$  small cycles:  $\mathcal{G} = \{\Gamma_1, \dots, \Gamma_k\}$ .



**Fig. 5.** From the graph created by an incremental set of latticelet’s arcs, small repetitive cycles  $\Gamma$  are selected by using a Breadth-first Search algorithm. Chains are created on the remaining nodes that have not been satisfied by any polygonal cycles  $\mathcal{G}$ .

In addition to what has been explained above, the algorithm tries to represent with chains the nodes that cannot be described with polygonal cycles. The procedure is analogous to the former one: chain arcs are selected by using the sorted latticelet set. The procedure is run for each object category.

## 6 Structure Inference Using Conditional Random Fields

So far, we showed our method to detect objects represented as closed contours and to find repetitive patterns in the occurrence of such objects. However, in many cases, objects can not be detected due to occlusions or low contrast in the image. In general, the problem of these false negative detections can not be solved, as there is not enough evidence of the occurrence of an object. In our case, we can use the additional knowledge that similar objects have been detected in the same scene and that all objects of the same kind are grouped according to a repetitive pattern. Using these two sources of information, we can infer the existence of an object, even if its detection quality is very low. We achieve this by using a probabilistic model: each possible location of an object of a given category  $\tau$  is represented as a binary random variable  $l_\tau(\mathbf{x})$  which is true if an object of category  $\tau$  occurs at position  $\mathbf{x}$  and false otherwise. In general, the state of these random variables can not be observed, i.e. they are *hidden*, but we can observe a set of features  $\mathbf{z}(\mathbf{x})$  at the given position  $\mathbf{x}$ . The features  $\mathbf{z}$  here correspond to the detection quality defined in Eqn. (II). The idea now is to find states of all binary variables  $\mathbf{l}_\tau = \{l_\tau(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\}$  so that the likelihood  $p(\mathbf{l}_\tau \mid \mathbf{z})$  is maximized. In our formulation we will not only reflect the dependence between the variables  $l$  and  $\mathbf{z}$ , but also the *conditional dependence* between variables  $l_\tau(\mathbf{x}_1)$  and  $l_\tau(\mathbf{x}_2)$  given  $\mathbf{z}(\mathbf{x}_1)$  and  $\mathbf{z}(\mathbf{x}_2)$ , where  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are positions that are very close to each other. The intuition behind this is that the occurrence probability of an object at position  $\mathbf{x}_1$  is higher if the same object already occurred at position  $\mathbf{x}_2$ . We model this conditional dependence by expressing the overall likelihood  $p(\mathbf{l}_\tau \mid \mathbf{z})$  as a CRF.

### 6.1 Conditional Random Fields

A CRF is an undirected graphical model that represents the joint conditional probability of a set of hidden variables (in our case  $\mathbf{l}_\tau$ ) given a set of observations  $\mathbf{z}$ . A node in

the graph represents a hidden variable, and an edge between two nodes reflects the conditional dependence of the two adjacent variables. To compute  $p(\mathbf{l}_\tau | \mathbf{z})$ , we define *node potentials*  $\varphi$  and *edge potentials*  $\psi$  as

$$\varphi(\mathbf{z}_i, l_{\tau i}) = e^{\mathbf{w}_n \cdot \mathbf{f}_n(\mathbf{z}_i, l_{\tau i})} \quad \text{and} \quad \psi(\mathbf{z}_i, \mathbf{z}_j, y_i, y_j) = e^{\mathbf{w}_e \cdot \mathbf{f}_e(\mathbf{z}_i, \mathbf{z}_j, l_{\tau i}, l_{\tau j})}, \quad (2)$$

where  $\mathbf{f}_n$  and  $\mathbf{f}_e$  are feature functions for the nodes and the edges in the graph (see below), and  $\mathbf{w}_n$  and  $\mathbf{w}_e$  are the feature weights that are determined in a training phase from hand-labeled training data. Using this, the overall likelihood is computed as

$$p(\mathbf{l}_\tau | \mathbf{z}) = \frac{1}{Z(\mathbf{z})} \prod_{i=1}^N \varphi(\mathbf{z}_i, l_{\tau i}) \prod_{(i,j) \in \mathcal{E}} \psi(\mathbf{z}_i, \mathbf{z}_j, l_{\tau i}, l_{\tau j}), \quad (3)$$

where  $Z$  is the *partition function*,  $N$  the number of nodes, and  $\mathcal{E}$  the set of edges in the graph. The computation of the partition function  $Z$  is intractable due to the exponential number of possible states  $\mathbf{l}_\tau$ . Instead, we compute the *log-pseudo-likelihood*, which approximates  $\log p(\mathbf{l}_\tau | \mathbf{z})$ .

In the training phase, we compute the weights  $\mathbf{w}_n$  and  $\mathbf{w}_e$  that minimize the negative log pseudo-likelihood together with a Gaussian shrinkage prior. In our implementation, we use the Fletcher-Reeves method [19]. Once the weights are obtained, they are used in the detection phase to find the  $\mathbf{l}_\tau$  that maximizes Eq. (3). Here, we do not need to compute the partition function  $Z$ , as it is not dependent on  $\mathbf{l}_\tau$ . We use max-product loopy belief propagation [20] to find the distributions of each  $l_{\tau i}$ . The final classification is then obtained as the one that is maximal at each node.

## 6.2 Node and Edge Features

As mentioned above, the features in our case are directly related to the detection quality obtained from Eqn. (1). In particular, we define the node features as  $\mathbf{f}_n(q_i, l_{\tau i}) = 1 - l_{\tau i} + (2l_{\tau i} - 1)q_i$ , i.e. if the label  $l_{\tau i}$  is 1 for a detected object, we use its detection quality  $q_i$ , otherwise we use  $1 - q_i$ . The edge feature function  $\mathbf{f}_e$  computes a two-dimensional vector as follows:

$$\mathbf{f}_e(q_i, q_j, l_{\tau i}, l_{\tau j}) = \begin{cases} \frac{1}{\gamma} \begin{pmatrix} f_{e1} & f_{e2} \\ 0 & 0 \end{pmatrix} & \text{if } l_{\tau i} = l_{\tau j} \\ \begin{pmatrix} 0 & 0 \end{pmatrix} & \text{else} \end{cases} \quad \text{with} \quad \begin{cases} f_{e1} = \max(\mathbf{f}_n(q_i, l_{\tau i}), \mathbf{f}_n(q_j, l_{\tau j})) \\ f_{e2} = \max_{G \in \mathcal{G}_{ij}} (\mathbf{f}_n(\eta(G), l_{\tau i})), \end{cases}$$

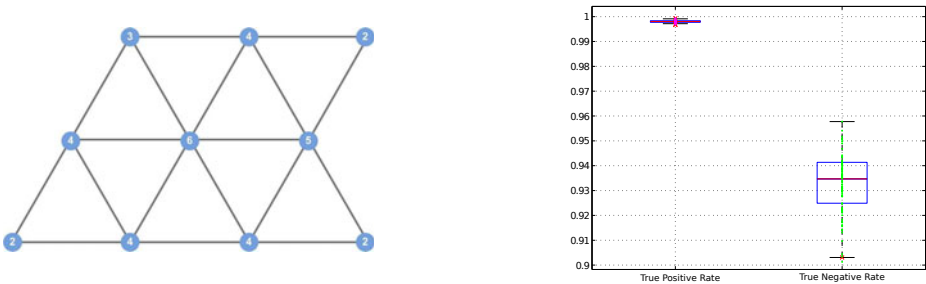
where  $\mathcal{G}_{ij}$  is the set of (maximal two) minimum cycles  $\Gamma$  that contain the edge between nodes  $i$  and  $j$ , and  $\eta(\Gamma)$  is a function that counts the number of detected objects along the cycle  $\Gamma$ , i.e. for which the detection quality is above  $\theta_q$ .

## 6.3 Network Structure

The standard way to apply CRFs to our problem would consist in collecting a large training data set where all objects are labeled by hand and for each object category  $\tau$  a pair of node and edge features is learned so that  $p(\mathbf{l}_\tau | \mathbf{z})$  is maximized. However, this approach has two major drawbacks:

- For a given object category  $\tau$ , there are different kinds of lattice structures in which the objects may appear in the training data. This means that the connectivity of a given object inside its network varies over the training examples. Thus, the importance of the edges over the nodes can not be estimated in a meaningful way.
- In such a supervised learning approach, only objects of categories that are present in the training data can be detected. I.e., if the CRF is trained only on, say, some different kinds of windows, it will be impossible to detect other kinds of objects that might occur in repetitive patterns in a scene. Our goal however, is to be independent of the object category itself and to infer only the structure of the network. In fact, the object category is already determined by the similarity detection described above.

To address these issues, we propose a different approach. Considering the fact that from the training phase we only obtain a set of node and edge weights  $\mathbf{w}_n$  and  $\mathbf{w}_e$ , which do not depend on the network geometry but only on its topology, we can artificially generate training instances by setting up networks with a given topology and assigning combinations of low and high detection qualities  $q_i$  to the nodes. The advantage of this is that we can create a higher variability of possible situations than seen in real data and thus obtain a higher generalization of the algorithm. The topology we use for training has a girth  $\gamma$  of 3 and is shown in Fig. 6 on the left. Other topologies are possible for training, e.g. using squared or hexagonal cycles, but from experiments we carried out it turns out that the use of such topologies does not increase the classification result. The graph in Fig. 6 right illustrates that. It shows the true positive and the true negative rates from an experiment with 100 test data sets, each consisting of networks with a total of 5000 to 10000 nodes. The training was done once only with a triangular topology (TriTop) and once also including square and hexagonal topologies (MixTop), which represent all possible regular tessellations of the plane. As the graph shows, there is no significant difference in the two classification results. In contrast to the topology, the number of outgoing edges per node, i.e. the *connectivity*, has a strong influence on the



**Fig. 6. Left:** Triangular lattice topology used for training the CRF. The numbers inside the nodes show the connectivity of the nodes. **Right:** Comparison of CRF performances using TriTop and MixTop datasets for training. True positive and the true negative rates are evaluated. The result from the TriTop data are shown in box-and-whiskers mode, the MixTop result as dots. We can see that using different topologies for learning gives no significant change in the classification result.

learned weights. Thus, we use a training instance where all possible connectivities from 2 to 6 are considered, as shown in Fig. 6 left.

In the inference phase, we create a CRF by growing an initial network. From the analysis of repetitive patterns described above, we obtain the set  $\mathcal{G}$  for each category, the topology and edge lengths of the lattice. By subsequently adding cycles from  $\mathcal{G}$  to the initial network obtained from the already detected objects, we grow the network beyond its current borders. After each growing step, we run loopy belief propagation to infer the occurrence of objects with low detection quality. The growth of the network is stopped as soon as no new objects are detected in any of the 4 directions from the last inference steps.

## 7 Model Compression

One aim of our work is to show that the information contained in an image (e.g. a facade) can be compressed using the proposed repetition detection technique. We reduce the image to a simple set of detected object categories, their repetition scheme, and a simplified background extraction. More in detail: each object category is stored as a set of codebook descriptors and vote vectors, a rectangular colorscale bitmap resulting from averaging the image areas inside the detected elements bounding boxes. To visually simplify the image background, we assume that the space between detected elements in a category is covered by textures of the same kind. We sort object categories by their cardinality. Then, as a texture simplification, we compute the median color between the elements by sampling squared image patches. This color is assigned to a rectangle patch that extends from top to the bottom of each category. We iterate this procedure until all the image is covered. Missing empty spaces are filled with the color of the most populous group. Some examples are shown in the right part of Fig. 9.

An image compressed with our method can be used in a number of applications such as visual based localization, in which information is extracted only from the repeated pattern, or low-bitrate storage for embedded systems (e.g. UAV) that have to store/transmit large urban environments. In a more general fashion we consider that our approach should be useful in all those cases where the main goal is to identify places where repetitive patterns are present, although it is not as well suited to provide detailed reconstructions of the represented objects.

## 8 Experiments

The goal of our experimental evaluation is to investigate to which extent the proposed algorithm is capable to detect different categories of objects, to detect repetition rules and to run inference based on that information.

In order to obtain rich statistics on a wide range of object categories we prepared an image evaluation set composed of high contrast polygons at different sizes. 150 pictures of  $450 \times 150$  pixels size have been computer generated, each one containing 2 to 8 different object categories. An object category is defined by a type of a polygon. It is important to stress that such set evaluates not the detection capabilities but the capacity of grouping similar elements, detecting latticelets and inferring high level cycles and

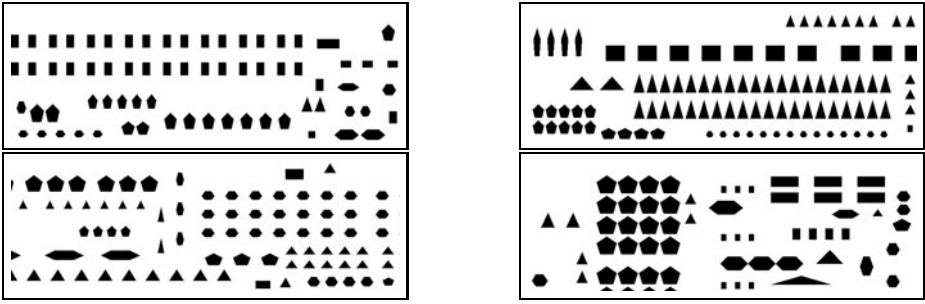


Fig. 7. Samples from the evaluation data set

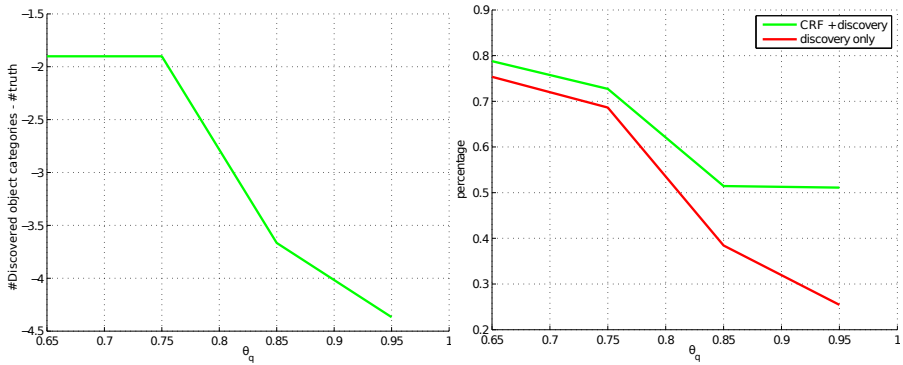
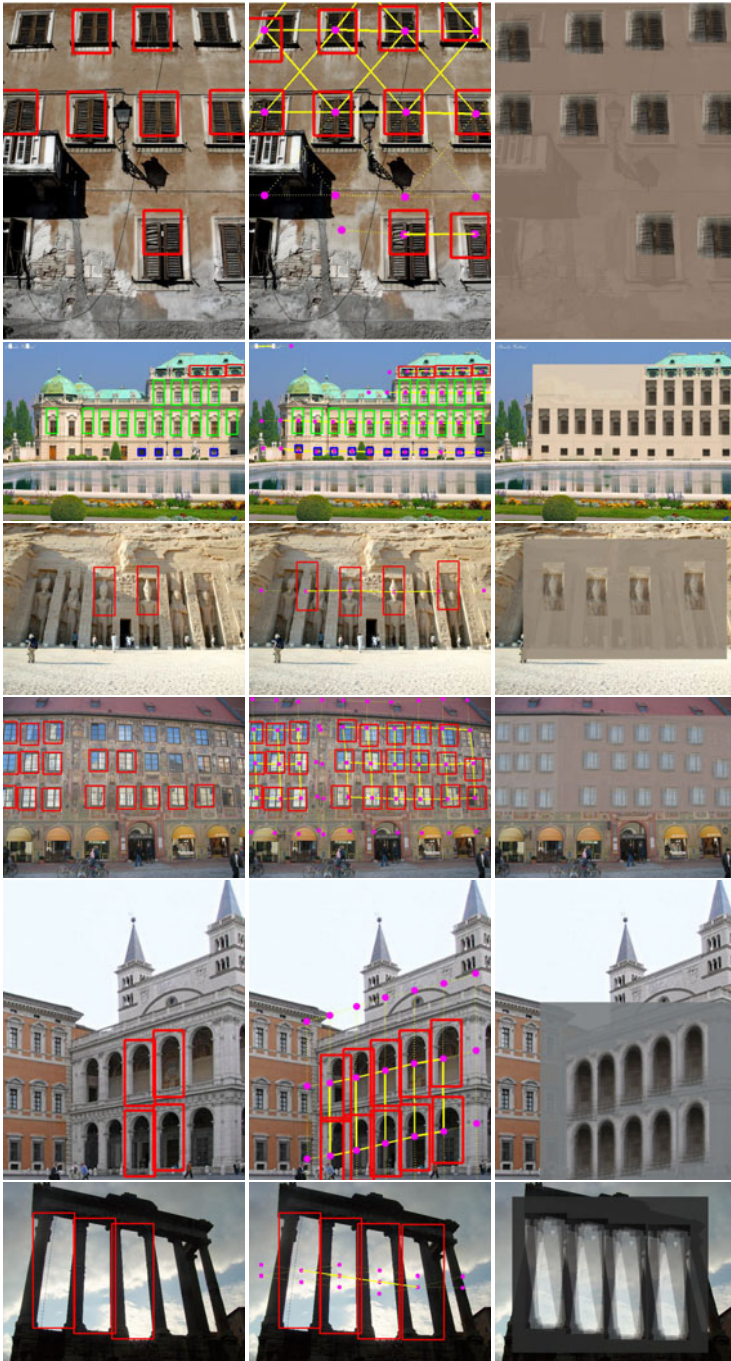


Fig. 8. **Left:** Average difference between the number of detected categories and annotated categories. The algorithm tends to under-explain the data trying to not overfit single detections. **Right:** Discovery only detection and discovery + CRF detection. The contribution of CRF for detecting missing elements is particularly evident when a low detection rate is obtained. Graphs are plotted with respect to the minimum detection quality  $\theta_b$  needed for each node.

chains for model compression and completion. Polygons are described by few pixels to introduce ambiguity in the description of repetitive elements. Fig. 7 shows some samples from the evaluation dataset.

One of our goals is to assess the quality of object category distinction and grouping, that is fundamental for the creation of the graph, as well as its analysis. It is important to note that the angle difference between an hexagon and a pentagon is just  $12^\circ$  and in small scales, due to pixel aliasing, this difference may not be easy to distinguish. Fig. 8 left shows the average difference between the number of detected categories and annotated categories. The graph is plotted with respect to the minimum detection quality  $\theta_b$  needed for each node. We can notice that the algorithm tends to under-explain the data trying to not overfit single detections. This is the result of the soft detection and grouping strategy we use that favors the merging of similar categories to the creation of a new one.

Moreover, we evaluate the contribution of the CRF to the detection rate of repetitive elements present in the image. We plot, in Fig. 8 right, this measure with respect to  $\theta_b$



**Fig. 9.** **Left Column:** Extracted self-similar objects (red boxes). Note that often only a few number of instances are found. **Center Column:** Final CRF lattice (dots and lines) and inferred position of objects (boxes). **Right Column:** Reconstruction of images based on our model compression.



and we overlay the results using CRF. The left side of the graph shows the CRF contribution (4%) when many annotated objects have been already detected by the discovery process, the right one shows the performance when just few elements are detected. In the latter case, a sound 20% detection rate improvement is achieved: it suffices that a small group of elements is detected for generating a set of  $\mathcal{G}$  used for inferring many missing aligned low-detection nodes. Important to mention is the average of false positives per image: 0.2. CRF therefore increases the true positive rate and it guarantees a very low false positive rate.

We also performed a quantitative analysis of compression ratio for the images in the evaluation set and the real-world images displayed in Fig. 9-right. The resulting compressed image is very compact and it stores just one bitmap for each object category and a list of 2D coordinates of elements locations. If we employ the ratio in bytes between the compressed image and the raw input image for the testing set images we obtain 1.4% ratio, for the pictures displayed in Fig. 9 (top to bottom order), we obtain: 2%, 1.2%, 2.3%, 0.8%, 2.8%, 8%. Even though this method aggressively reduces the amount of image details, the salient repetitive pattern is preserved.

A set of images of facades and other repetitive elements have been downloaded from internet and treated as input for our algorithm, Fig. 9. On each of the examples the difference from discovery and CRF-completed image is shown. It is interesting to notice that the algorithm works also for not rectified facades and several kind of architectural or repetitive elements. In the scope of this work it is evident that training on a simulated data, sufficiently rich in variability, satisfies also real world examples.

## 9 Conclusions

In this paper we presented a probabilistic technique to discover and reason about repetitive patterns of objects in a single image. We introduced the concepts of latticelets, generalized building blocks of repetitive patterns. For high-level inference on the patterns, CRFs are used to soundly couple low-level detections with high-level model information.

The method has been tested on simulated and real data showing the effectiveness of the approach. From a set of synthetic images, it was verified that the method is able to correctly learn different object categories in an unsupervised fashion regardless the detection thresholds. For the task of object detection by model prediction and completion, the experiments showed that the method is able to significantly improve detection rate by reinforcing weak detection hypotheses with the high-level model information from the repetitive pattern. This is especially true for large thresholds for which detection only, without our method, tends to break down. For the task of model compression, i.e. retaining and efficiently representing the discovered repetitive patterns, a very high compression ratio of up to 98% with respect to the raw image has been achieved.

Beyond the tasks of model completion and compression, we see applications of this method in image inpainting, environment modeling of urban scenes and robot navigation in man-made buildings.

## Acknowledgements

This work has been partly supported by German Research Foundation (DFG) under contract number SFB/TR-8 and EU Project EUROPA-FP7-231888.

## References

1. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: ECCV Workshop on Stat. Learn. in Comp. Vis. (2004)
2. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmentation and labeling sequence data. In: Int. Conf. on Mach. Learn. (ICML) (2001)
3. Dick, A.R., Torr, P.H.S., Cipolla, R.: Modelling and interpretation of architecture from several images. *Int. Journ. of Comp. Vis.* 60, 111–134 (2004)
4. Mayer, H., Reznik, S.: Building facade interpretation from image sequences. In: ISPRS Workshop CMRT 2005 (2005)
5. Xiao, J., Fang, T., Tan, P., Zhao, P., Ofek, E., Quan, L.: Image-based façade modeling. In: ACM SIGGRAPH Asia (2008)
6. Wonka, P., Wimmer, M., Sillion, F., Ribarsky, W.: Instant architecture. *ACM Trans. Graph.* 22, 669–677 (2003)
7. Müller, P., Zeng, G., Wonka, P., Van Gool, L.: Image-based procedural modeling of facades. *ACM Trans. Graph.* 26, 85 (2007)
8. Schaffalitzky, F., Zisserman, A.: Geometric grouping of repeated elements within images. In: British Machine Vision Conf. (1999)
9. Turina, A., Tuytelaars, T., Van Gool, L.: Efficient grouping under perspective skew. In: IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR) (2001)
10. Liu, Y., Collins, R.T., Tsin, Y.: A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE Trans. Pattern An. & Mach. Intell.* 26, 354–371 (2004)
11. Hays, J., Leordeanu, M., Efros, A.A., Liu, Y.: Discovering texture regularity as a higher-order correspondence problem. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 522–535. Springer, Heidelberg (2006)
12. Korah, T., Rasmussen, C.: Analysis of building textures for reconstructing partially occluded facades. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 359–372. Springer, Heidelberg (2008)
13. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern An. & Mach. Intell.* 24, 509–522 (2002)
14. Canny, J.: A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 8, 679–698 (1986)
15. Freeman, H.: On the encoding of arbitrary geometric configurations. *IEEE Trans. Electr. Computers EC-10*, 260–268 (1961)
16. Spinello, L., Triebel, R., Siegwart, R.: Multimodal people detection and tracking in crowded scenes. In: Proc. of the AAAI Conf. on Artificial Intelligence (2008)
17. Kruskal, J.B.: On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society* 7, 48–50 (1956)
18. Collins, R., Beveridge, J.: Matching persp. views of copl. structures using projective unwarping and sim. matching. In: IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR) (1993)
19. Fletcher, R.: *Practical Methods of Optimization*. Wiley, Chichester (1987)
20. Pearl, J.: *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco (1988)

# Feature Tracking for Wide-Baseline Image Retrieval

Ameesh Makadia

Google Research,  
76 Ninth Ave,  
New York, NY 10014  
makadia@google.com

**Abstract.** We address the problem of large scale image retrieval in a wide-baseline setting, where for any query image all the matching database images will come from very different viewpoints. In such settings traditional bag-of-visual-words approaches are not equipped to handle the significant feature descriptor transformations that occur under large camera motions. In this paper we present a novel approach that includes an offline step of feature matching which allows us to observe how local descriptors transform under large camera motions. These observations are encoded in a graph in the quantized feature space. This graph can be used directly within a soft-assignment feature quantization scheme for image retrieval.

**Keywords:** Wide baseline, image retrieval, quantization.

## 1 Introduction

In this paper we address the challenge of image retrieval from large databases. While this is a classic problem in Computer Vision, we are interested in the specific scenario of *wide-baseline* image retrieval. In this setting, we assume that for most query images the closest matching (true matches) databases images are of the same scene but from very different viewpoints, and thus have undergone significant transformations relative to the query (see Figure 5 for an example).

We isolate the wide-baseline challenge because it has important practical implications. For example, it is unrealistic in any real-world image retrieval system that the databases will contain many images of all interesting locations, so matching to a few images of a scene is important. Furthermore, the ability to effectively match images from a wide-baseline means one can construct the database accordingly, keeping fewer images of each scene than would otherwise be needed. This would have an impact on both storage costs and retrieval time.

Much of the work for large scale image retrieval has been based on the bag-of-visual-words (BOW) approach [1,2], which borrows ideas from the text-retrieval community. To summarize briefly, forming an image representation can be broken into three steps: (1) local feature detection and extraction (2) feature quantization, and (3) tf-idf image representation. In our setting, the real challenges lie

within the first two steps. Much effort has been put into identifying local feature detectors and descriptors [3,4,5] suitable for correspondence problems, but even these will not remain sufficiently invariant (either repeatability of detector or invariance of descriptor) when the camera undergoes very large motions. Also, while feature quantization may mask small descriptor transformations, it is unlikely to deal gracefully with larger transformations.

In this paper we aim to improve wide-baseline retrieval within the BOW framework. To address the primary issue of feature deformations over large camera motions, we perform unsupervised tracking of millions of points through long image sequences in order to observe how corresponding descriptors transform under significant camera motions. As an intermediate representation for millions of feature tracks, we construct a weighted graph embedded in the quantized feature space, where the edge weight between two words is related to the number of tracks having descriptors mapped to both words. We will refer to this graph as the *track-graph*. In a way the track-graph encodes how often we have seen features that are mapped to one word transform into features that are mapped to another word. Importantly, this graph construction provides a purely data-driven way of encoding the observed feature transformations. We avoid the difficult problem of explicitly modeling or parameterizing the space of feature transformations, for example. We utilize the track-graph for the image retrieval application by incorporating it into a soft-assignment scheme similar to that of [6].

Our primary contribution can be summarized as a novel approach for image retrieval that utilizes offline feature tracking to observe feature transformations under large camera motions. To our knowledge this is the first method that successfully incorporates such information within a BOW retrieval approach. Furthermore, we examine properties of the track-graph in detail to understand the added value of the information it encodes. Evaluation of the retrieval system in a wide-baseline setting shows promising performance.

## 1.1 Related Work

Retrieval from large image collections is a well-studied problem in Computer Vision. In 2003, Sivic and Zisserman [1] applied a text retrieval approach for object retrieval, and many of the current state-of-the-art methods can be seen as an extension of this BOW approach (a few examples are [2,7,8,9]).

At their core, these methods rely on a quantization of the feature space into visual words to make large scale retrieval a tractable problem. To this end a number of papers have explored various ideas related to partitioning and assignment in the feature space. The baseline standard is  $k$ -means clustering to build a vocabulary, and nearest-neighbor assignment of descriptors to words [1]. In [2], a vocabulary tree (constructed with hierarchical  $k$ -means) is used for feature quantization and assignment, while [8] considers approximate  $k$ -means for assignment, and [10] considers a fixed quantization for a vocabulary. More recently, [7] have studied the effects of quantization and introduce a combination of vector quantization and hamming embedding. In [11] kernel density estimation

is applied to capture the uncertainty of assignment from features to words, thus limiting the effects of hard assignment. Philbin et al [6] show the effects of quantization can be remedied in part by a soft-assignment when mapping descriptors to words. Our work is influenced by this last approach as we will incorporate our observations of feature transformations into a similar soft-assignment scheme. While the works above address the issue of feature quantization and assignment, there has been little done to specifically address the challenges of image retrieval under wide-baseline conditions. Efforts in landmark recognition are capable of building rich 3D representations of landmarks or scenes given a sufficient number of images [12,13,14,15]. These representations can be utilized for image matching and retrieval. However, such approaches [13] still require the image database to be populated with a large number of images of each landmark. This is in contrast to our wide-baseline setting where we do not expect to have a large number of matching images in the database for any query.

A central premise of our approach is that by tracking or matching features through image or video sequences one can observe how image features transform under large viewpoint changes. Utilizing this information should improve retrieval when query images come from unique viewpoints relative to the database images. Prior related work includes [16], where Implicit Shape Models are constructed for object pose recognition. 3D descriptors are represented by a set of 2D image descriptors that were matched over a set of training images, however manual correspondences (e.g. hand-selected feature matches) are required for initialization. In [17], object level representations are formed by object regions tracked within video shots, which provides viewpoint invariance for retrieval. In [18], feature matching between images in the training set is used to identify a set of “useful” features. This greatly reduces the number of features stored from each image without loss in retrieval precision. Earlier work [19] explores building invariant distance measures with a priori knowledge of patch transformations. In [20] invariance is incorporated into SVM classifiers by introducing artificially transformed copies of support vectors. In [21], feature matching under wide-baseline conditions is treated as a classification problem, where each class corresponds to all views of a point. In the BOW approach of [6], one variation of their soft assignment scheme involves generating multiple feature sets for an image by applying small transformations directly to the image. While our work is related to these last approaches, we note that in both [21] and [6] feature transformations are generated by *simulating* image transformations. This is limited in that one has to model and parameterize the space of patch transformations (e.g. affine transformations), and it is unlikely such an approach can capture the full spectrum of transformations that are observed from actual camera motions. In our work we address this issue by tracking features through image sequences.

## 2 Bag-of-Visual-Words Image Retrieval

In this section we give a brief summary of our implementation of the traditional baseline BOW model. We combine a Hessian-Affine interest point detector and

SIFT descriptors [45], and starting with a collection of SIFT features from a large dataset a visual vocabulary is constructed by partitioning the feature space with  $k$ -means clustering. The ideal number of clusters depends on the dataset, and so in this paper we experiment with vocabularies ranging from 2500 to 500000 in size. We denote a vocabulary as  $V = \{v_1, v_2, v_3, \dots\}$ , where the visual words  $v_i$  are the cluster centers. Feature-to-word mapping assigns a descriptor  $x_i$  to the closest visual word  $\hat{x}_i = \underset{v}{\operatorname{argmin}} \|x_i - v\|_2$ . In practice, this mapping is done with approximate nearest neighbors when the vocabulary size is large. The final tf-idf image representation is simply a weighted histogram over the words appearing in an image. For complete details see [122].

### 3 Constructing the *Track-Graph*

In this section we describe our process of constructing a graph in the quantized space that encodes feature transformations observed from image sequences. Our intermediate goal here is to develop a method that allows us to characterize how feature descriptors transform under large camera motions. As with any data-driven approach, we must be sure to collect a large number of samples in order to be sure our observations have statistical significance. Our setup consists of 3 cameras on top of a vehicle that acquires images while driving through urban city streets. The three cameras ( $c_1$ ,  $c_2$ , and  $c_3$ ) are arranged to face the same side of the street, but at different angles. Camera  $c_2$  is fronto-parallel to building facades on the side of the street, while  $c_1$  and  $c_3$  are offset approximately  $45^\circ$  on either side of  $c_2$ . Due to this arrangement, scene content that is observed by camera  $c_1$  at time  $t$  is usually observed by  $c_2$  and  $c_3$  at a later time  $t' > t$ . Figure 1 shows a sequence of images from 7 consecutive time steps.



**Fig. 1.** A sequence of seven consecutive frames from our acquisition system that illustrates how the same scene content will be seen from multiple cameras over time separated by a wide baseline. At each of the seven time steps (frames 1116 through 1122), the same scene content is observed from one of the three cameras (1, 2, or 3). With frame-to-frame feature matching we are more likely to generate tracks that observe world points from very different viewpoints angles than if we tried to match features directly between the extreme wide baseline image pairs.

### 3.1 Feature Matching and Track Extraction

Since the sampling rate of the camera is too low to allow true feature tracking between frames, we pursue discrete matching with RANSAC [23] to generate correspondences that are consistent with a Fundamental matrix or Homography transformation. Putative correspondences are obtained with nearest-neighbor matching while discarding ambiguous candidates [5] (however all nearest neighbor-matches are used to generate the final set of inliers after geometry estimation). Since image acquisition is of minor cost, our matching thresholds are fairly strict to protect against too many outliers. In total, we collected five non-overlapping image sequences from the same city that contained a total of 45K image frames. Our matcher extracted 3.8M feature tracks having an average duration of 5.8 frames (tracks shorter than 3 frames are discarded).<sup>1</sup> Employing discrete matching in place of pure tracking also serves a secondary purpose. It is well known that repeatability of a feature detector is limited under viewpoint changes, and so by generating tracks in this way we make sure to observe feature transformations only for those features where the reliability of the detector is certain.

### 3.2 Graph Construction

We define the *track-graph* as a weighted graph  $G = (V, E, w)$ , where the vertices are the set of visual words in the vocabulary. We would like the weight  $w(u, v)$  to reflect the number of feature tracks whose descriptors have mapped to both  $u$  and  $v$ . Let us represent a tracked feature  $t$  with its observed SIFT descriptors  $t = \{x_1, x_2, x_3, \dots\}$ , and let  $T$  be the collection of all feature tracks obtained during the offline tracking process ( $T = \{t_i\}$ ). Figure 2 shows the few steps required in constructing the weighted graph. To summarize, the edge weights between vertices  $w(u, v)$  count exactly the number of tracks where at least one descriptor mapped to word  $u$  and at least one descriptor mapped to word  $v$ . Note that our construction process ignores self-edges ( $w(u, u) = 0, \forall u \in V$ ).<sup>2</sup>

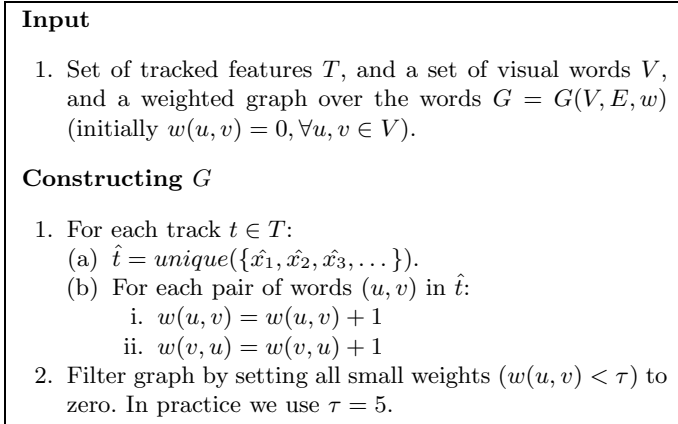
### 3.3 Properties of the Track-Graph

The graph construction process described above can be seen as an iterative process, where tracks are incorporated into the track-graph one after another (in arbitrary order). The natural question that arises is how do we determine the stability of the graph as we continue to incorporate more tracks (this can help us determine if we have included enough observations to terminate construction). To study this characteristic we evaluate how the graph changes as we continue

<sup>1</sup> Figures depicting some tracking results can be seen at

<http://www.cis.upenn.edu/~makadia/>

<sup>2</sup> We also considered post-processing the graph to account for the frequency in which certain types of features were tracked. One such normalization was, for example,  $w'(u, v) = \frac{w(u, v)^2}{\sum_y w(u, y) \sum_y w(v, y)}$ . In practice, the few we tried all lowered performance.



**Fig. 2.** Outline of track-graph construction. Here a track  $t$  is represented by the observed feature descriptors  $t = \{x_1, x_2, \dots\}$ , and the notation  $\hat{x}_i$  refers to the visual word assignment for feature  $x_i$ .

to add more tracks. Given a graph  $G_1$ , let us define the probability of seeing an edge  $(u, v)$  as  $P((u, v)) = \frac{w(u, v)}{\sum_{u, v \in V} w(u, v)}$ . Note for this task we ignore that  $w(u, v)$  and  $w(v, u)$  represent the same link. Given two graphs  $G_1$  and  $G_2$ , the KL-divergence of  $P_{G_2}$  from  $P_{G_1}$  is used to measure their relative difference:

$$D_{KL}(P_{G_2} \| P_{G_1}) = \sum_{(u, v) \in V \times V} P_{G_2}((u, v)) \log \frac{P_{G_2}((u, v))}{P_{G_1}((u, v))} \quad (1)$$

For our purposes here  $G_2$  will always represent a graph obtained by integrating more tracks into  $G_1$ . The relative graph distance is not complete unless we account for the relative “sizes” of the graphs. In other words, the relative change in graphs should be normalized by the number of observations used to construct the graphs. If we define the size of the graph as  $W_G = \sum_{u, v \in V} w(u, v)$ , we can define the relative change between graphs as the KL-divergence scaled by the relative change in graph size:  $D(P_{G_2} \| P_{G_1}) = D_{KL}(P_{G_2} \| P_{G_1}) \frac{W_{G_1}}{W_{G_2}}$ . Table 1 shows that the graph changes much less as more and more tracks are incorporated (in this example the vocabulary size is 250000 words). This experiment was performed on graphs before the small edges were filtered out (as per Figure 2), which means the stability is observed even with possibly noisy edges present. The second important consideration is whether or not the constructed track-graph contains information that cannot be obtained from standard distance measures in the feature space. If after constructing a graph from millions of tracked features we find that the nearest neighbors in the graph (according to the edge weights) mimic the neighbors produced with a traditional distance measure, this would indicate that the track-graph will not contribute any orthogonal information. To examine this property, we construct another graph that captures the proximity



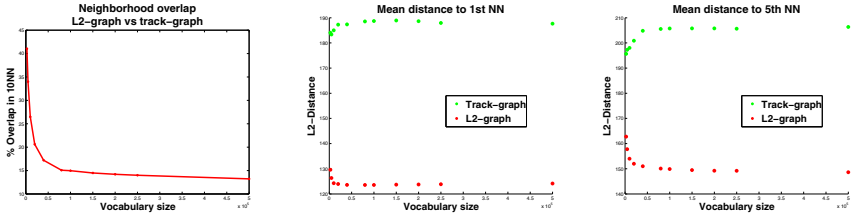
**Table 1.** Each column shows how the track-graph changes as a new collection of tracks is incorporated. See the text for term definitions. The last row indicates that as we incorporate more and more tracks into the graph, the relative change in the graph continues to decrease. Note, in each column  $G_2$  represents a graph obtained after incorporating more tracks into  $G_1$ .

$W_{G_1}$	12M	20M	29M	36M
$W_{G_2}$	20M	29M	36M	41M
$\frac{W_{G_1}}{W_{G_2}}$	0.58	0.70	0.79	0.88
$D_{KL}(P_{G_2}  P_{G_1})$	0.36	0.20	0.13	0.07
$D(P_{G_2}  P_{G_1})$	<b>0.21</b>	<b>0.14</b>	<b>0.10</b>	<b>0.06</b>

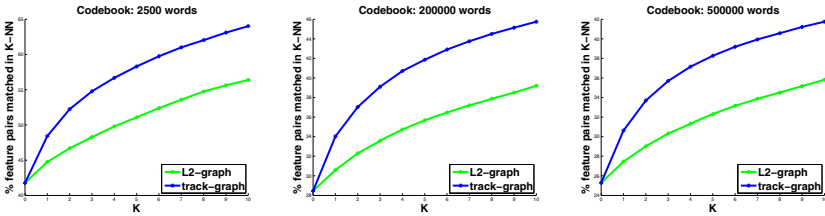
between visual words in the feature space using a standard distance measure for SIFT features. We call this the  $L_2$ -graph since the edge weight  $w(u, v)$  is related to the Euclidean distance between  $u$  and  $v$ . To see how the track-graph and  $L_2$ -graph relate, we compare for each visual word its closest neighbors in the track graph against its closest neighbors in the  $L_2$ -graph. Figure 3 (left) illustrates the average overlap between a visual word’s 10 closest neighbors in the two graphs. Even for small vocabulary sizes there is less than 50% overlap between the neighborhoods. A related experiment shown in Figure 3 (middle, right) examines the actual Euclidean distance to a word’s  $k$ -th nearest neighbor in the track and  $L_2$  graphs, respectively. The differences between the two graphs is an indication that the track graph is capturing feature transformations that may not occur smoothly in the feature space.

A final experiment on the graphs is a simple test of feature assignment. The idea is to see how useful the track-graph weights might be in practice where corresponding features are often initially mapped to different words. Since our graph weights are constructed to reflect this property exactly, in some sense this test can be considered a cross-validation step. We collect feature tracks from an image sequence that was not used during track-graph construction. From these tracks we select 5000 wide-baseline feature pairs. We consider a feature pair  $(x_i, x_j)$  to be wide-baseline if  $x_i$  was observed in camera  $c_1$  and  $x_j$  in  $c_3$  (or vice-versa). Our measure of correct assignment is if  $\hat{x}_j$  is one of the  $k$ -nearest neighbors of  $\hat{x}_i$  in the track/ $L_2$  graph. Figure 4 shows the results of this experiment for different vocabulary sizes and for  $k$  ranging from 0 to 10 ( $k = 0$  is just traditional hard assignment, and is thus the same for both the track and  $L_2$  graphs).

The experiments above depict valuable qualities of the track-graph. First, the graph is relatively stable after incorporating 3.8M tracks (Table 1), which gives us confidence we have included enough tracks during construction. Second, the graph encodes some information about how feature descriptors transform that cannot be observed with a traditional distance measure (Figure 3). Finally, initial experiments show the graph may be useful in practical correspondence tasks (Figure 4).



**Fig. 3.** The single plot on the left compares the overlap between a visual word’s 10 nearest neighbors in the track-graph and its neighbors in the  $L_2$ -graph. We ignore those words that did not have any tracked neighbors. The plot shows this neighborhood overlap ratio for graphs constructed with 11 different vocabularies (ranging from 2500 to 500000 words). The two plots to the right compare the average distance of a visual word to its  $k$ -th nearest neighbor ( $k = 1$  on the left,  $k = 5$  on the right).



**Fig. 4.** Feature assignment results for the track and  $L_2$  graphs. Results are shown for graphs constructed with three different vocabulary sizes: 2500 (left), 200000 (middle), and 500000 (right). For each feature pair  $(x_i, x_j)$ , assignment is considered correct if  $x_j$  is one of the  $k$  nearest neighbors of  $x_i$  in the graph. In the plots  $k$  ranges between 0 and 10.

### 3.4 Image Retrieval with the Track-Graph

To utilize the track-graph in an image-retrieval engine, we develop a natural feature quantization scheme that is motivated by the soft assignment approach of Philbin et al [6]. To summarize [6] briefly, instead of quantizing feature  $x$  to its closest word  $\hat{x}$ , the vote for  $x$  in the tf-idf vector is distributed over the  $k$ -nearest words to  $x$ . The weights given to each of these words is proportional to  $\exp \frac{-d^2}{2\sigma^2}$ , where  $d = \|x - \hat{x}\|_2$ .

We utilize our track-graph in a similar way. Instead of assigning  $x$  to its closest word  $\hat{x}$  (determined by  $L_2$  distance), the vote for  $x$  will be distributed between  $\hat{x}$  and the closest words to  $\hat{x}$  in the track graph. For  $k$ -nearest neighbor assignment, for example, the weight for  $x$  will go to  $\hat{x}$  and the  $(k-1)$ -nearest neighbors of  $\hat{x}$  in the track-graph (neighbors are determined by sorting the edge weights  $w(\hat{x}, v)$  in decreasing order). Here also the tf-idf weights are proportional to  $\exp \frac{-d^2}{2\sigma^2}$ . The weights for each feature are scaled uniformly so the total tf-idf contribution is 1, and  $\sigma$  is set identical to [6]. Note, we are only using the graph weights  $w(\hat{x}, v)$  for selecting a word’s neighbors, while the tf-idf weights are determined by  $L_2$  distances. The fundamental difference between our approach and [6] is that the

track-graph provides a unique way of selecting the “closest” words to a feature. The track-graph neighbors will be consistent with the feature transformations observed in our offline tracking process rather than just  $L2$  proximity in the feature space. For visual words that have fewer than  $k - 1$  edges, the neighborhood is supplemented with the original feature’s closest ( $L2$ ) words. For example, if a feature’s closest word  $\hat{x}$  has no track-graph neighbors, its assignment reduces to the soft-assignment of [6]. For the track-graph constructed over 500000 visual words, 36% of the words had no edges. At the other extreme, all words had some neighbors for the graph constructed with the smallest vocabulary of 2500 words. In the next section we evaluate our proposed approach for wide-baseline image retrieval.

## 4 Evaluation

As we wish to evaluate image retrieval specifically in a wide-baseline setting, we prepared a test scenario that reflects the relevant challenges. We begin by collecting an image sequence in the same manner as described earlier. From this sequence, we select 1374 non-overlapping test images, and the remaining images form the database. All test images are chosen from either camera  $c_1$  or  $c_3$ , so that they are not oriented fronto-parallel to the building facades. To create a sufficiently difficult challenge, only the wide-baseline matches are stored in the database (6348 images). Figure 5 shows two test images and their true neighbors in the database. Each test image has on average 4.6 true matches in the database. We supplement the dataset with 247686 images collected from image sequences that have no overlap with the test sequence. In total, we have 1374 test images and 254034 database images.

We note that there is no overlap between the images used for vocabulary and graph construction and the images used to build the evaluation dataset. However, all images come from urban environments using the same image acquisition scheme so the vocabulary and tracked features will still be relevant for the application.



**Fig. 5.** Challenging sample test images and their closest matching database images. The significant camera motion between the queries and their matches makes for a difficult retrieval task.



**Fig. 6.** Top images retrieved for three queries using our track-graph method (Vocabulary size 500000, 5-NN assignment). The leftmost image (highlighted in red) is the query. The retrieval images are shown to the right of the query (correct matches highlighted in green). For all three queries, both [6] and traditional hard assignment failed to retrieve any correct matches in the top 10.

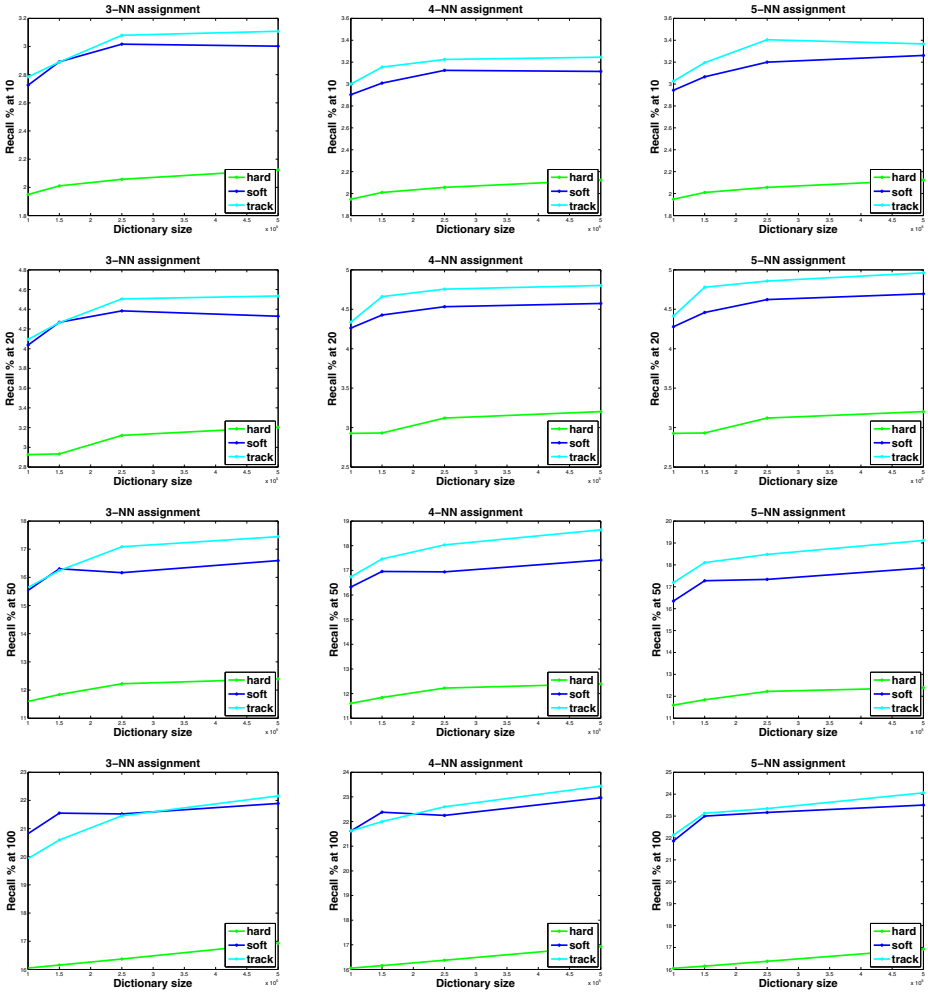
#### 4.1 Evaluation Criteria

Most image retrieval systems designed for practical large-scale use perform some form of post-processing or re-ranking of the top results from the initial retrieval (e.g. geometric verification, query expansion, see [8,24]). In this setting the most important criteria is making sure as many correct results as possible appear in the portion of the list that will be post-processed. In light of this we focus our evaluation on the top returned images. Specifically, we will measure recall (at  $n$ ), which measures what fraction of the true matches appear in the top  $n$  results. We will evaluate our track-graph based approach against the soft assignment of [6], as well as the traditional BOW approach.<sup>3</sup>

#### 4.2 Results

For our evaluation the two primary parameters are (1) the number of neighbors used in  $k$ -NN feature-to-word assignment and (2) the cutoff  $n$  for which we evaluate mean recall-at- $n$ . Figure 7 shows results for  $n \in \{10, 20, 50, 100\}$ , and  $k \in \{3, 4, 5\}$ . Of the 11 vocabulary sizes we have experimented with in previous sections (2500, 5000, 10000, 20000, 40000, 80000, 100000, 150000, 200000, 250000, and 500000), we select four of the larger vocabularies (100000, 150000, 250000, and 500000 words) for displaying results here (as expected, all three algorithms performed their best on these larger vocabularies). While both our track-graph approach and the soft assignment of [6] significantly outperform the traditional hard assignment, the results also show the track-graph consistently improving over [6], especially at the largest vocabularies. The improvements are

<sup>3</sup> We attempted a variation of soft assignment based on simulating image patch transformations [6], but due to the many implementation parameters our best results underperformed the simple BOW baseline, thus those results are not included here.



**Fig. 7.** Results of our track-graph approach ('track'), soft assignment [6] ('soft'), and traditional assignment ('hard'). Each plot shows a different combination of  $n$  (recall-at- $n$ ) and  $k$  ( $k$ -NN assignment that is used in both our approach as well as [6]).  $n$  is one of 10, 20, 50, or 100.  $k$  is one of 3, 4, or 5.

most noticeable at  $n = 50$ , while performance is closer at  $n = 100$ . For visual examples, Figure 6 shows three queries where our approach succeeded in retrieving at least one correct result in the top ten while both algorithms we compare against fail to return any correct matches.

Looking at these results in a wider context, the final improvement in the application setting of our method over the approach of [6] may seem modest compared to the possible gains indicated by our earlier isolated experiments (see Figure 4). One explanation for this is that, as mentioned earlier, our feature

mapping reverts to [6] for those words where we have no tracking observations. In the case of 500000 words we see that 36% of the words had no track-graph neighbors. Furthermore, the earlier experiments isolate the comparison of tracked neighbors and  $L2$  neighbors, whereas the retrieval results in Figure 6 show the results of an entire image retrieval engine, where naturally the differences within a single component will be muted.

Regarding the cost to sparsity using our track-graph approach, we note that for the 500000 word vocabulary, using  $3 - NN$  assignment, our approach generates 7% fewer nonzero entries in the tf-idf representation than the comparable [6] (while simple hard assignment produces 66% fewer nonzero entries). Another question is how does our constructed track-graph perform on image retrieval from more general image collections? We emphasize that it is critical that the track-graph encode the types of transformations expected in the retrieval problem (in this paper we focus on wide-baseline camera motions exhibited by street-level imagery). As we discuss in more detail in the following section, extending our automatic tracking and graph construction process to address the types of transformations observed in general collections (e.g. Web datasets) is non-trivial and left to future work<sup>4</sup>.

## 5 Future Work

We have designed a novel data-driven approach to study how image features transform under large camera motions and how such observations can be incorporated into a system targeting wide-baseline retrieval. While our results are promising, we consider this preliminary work and note a number of areas requiring future attention. Most notably is the generalization of our approach. While our current approach encodes wide-baseline (specifically planar) motions in the track-graph, going forward we would like to cover all possible transformations (e.g. descriptor transformations induced by general camera motions, lighting and environment changes, changes in camera modality, etc.). This extension is non-trivial because our approach requires simple data collection and fully unsupervised tracking to generate a large number of observations. However, extending this approach will be challenging because controlling data collection to observe a wide set of transformations, as well as automatically generating sufficient ground-truth correspondences, is not a simple task. Another point for future work is addressing the artifacts of quantization that remain in our development. Our graph construction relies on hard assignment of tracked descriptors to visual words, and similarly during tf-idf construction for identifying the word from which track neighbors are selected. While our decisions here have been motivated by sparsity and computation, we plan for future work to explore a fully probabilistic soft assignment framework for graph construction as well as

<sup>4</sup> However, as a validation of our intuition here we do provide an evaluation of our wide-baseline tracks applied out of context to a general Web dataset. The supplemental material at <http://www.cis.upenn.edu/~makadia/> shows performance on the Oxford Buildings dataset [8].

tf-idf generation. Another aspect of our work worth investigating further is the application of our ideas to different problems. For example, we believe the track-graph may be useful for improving correspondences in two views, and the offline feature tracking can be used to build better visual vocabularies.

**Acknowledgments.** We thank Matthew Burkhart and Alexander Toshev for helpful discussions, and the Google StreetView team for the data.

## References

1. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: International Conference on Computer Vision, pp. 1470–1477 (2003)
2. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2161–2168 (2006)
3. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: BMVC, pp. 383–393 (2002)
4. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. *Int. J. Comput. Vision* 60, 63–86 (2004)
5. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision* 60, 91–110 (2004)
6. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2008)
7. Jégou, H., Douze, M., Schmid, C.: Improving bag-of-features for large scale image search. *International Journal of Computer Vision* 87, 316–336 (2010)
8. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2007)
9. Jégou, H., Harzallah, H., Schmid, C.: A contextual dissimilarity measure for accurate and efficient image search. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2007)
10. Tuytelaars, T., Schmid, C.: Vector quantizing feature space with a regular lattice. In: International Conference on Computer Vision (2007)
11. Gemert, J.C., Geusebroek, J.M., Veenman, C.J., Smeulders, A.W.: Kernel codebooks for scene categorization. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5304, pp. 696–709. Springer, Heidelberg (2008)
12. Agarwal, S., Snavely, N., Simon, I., Seitz, S., Szeliski, R.: Building rome in a day. In: International Conference on Computer Vision (2009)
13. Li, X., Wu, C., Zach, C., Lazebnik, S., Frahm, J.M.: Modeling and recognition of landmark image collections using iconic scene graphs. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 427–440. Springer, Heidelberg (2008)
14. Zheng, Y.T., Zhao, M., Song, Y., Adam, H., Buddemeier, U., Bissacco, A., Brucher, F., Chua, T.S., Neven, H.: Tour the world: building a web-scale landmark recognition engine. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2009)
15. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3d. In: SIGGRAPH, pp. 835–846 (2006)

16. Arie-Nachimson, M., Basri, R.: Constructing implicit 3D shape models for pose estimation. In: International Conference on Computer Vision (2009)
17. Sivic, J., Schaffalitzky, F., Zisserman, A.: Object level grouping for video shots. *Int. J. Comput. Vision* 67, 189–210 (2006)
18. Turcot, P., Lowe, D.: Better matching with fewer features: The selection of useful features in large database recognition problems. In: ICCV Workshop on Emergent Issues in Large Amounts of Visual Data, Kyoto, Japan (2009)
19. Simard, P.Y., Cun, Y.A.L., Denker, J.S., Victorri, B.: Transformation invariance in pattern recognition - tangent distance and tangent propagation. LNCS, pp. 239–274. Springer, Heidelberg (1998)
20. Schölkopf, B., Burges, C., Vapnik, V.: Incorporating invariances in support vector learning machines. In: Vorbrüggen, J.C., von Seelen, W., Sendhoff, B. (eds.) ICANN 1996. LNCS, vol. 1112, pp. 47–52. Springer, Heidelberg (1996)
21. Lepetit, V., Pilet, J., Fua, P.: Point matching as a classification problem for fast and robust object pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 244–250 (2004)
22. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York (1986)
23. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 381–395 (1981)
24. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: International Conference on Computer Vision (2007)



# Crowd Detection with a Multiview Sampler

Weina Ge and Robert T. Collins

The Pennsylvania State University, University Park, PA 16802, USA

**Abstract.** We present a Bayesian approach for simultaneously estimating the number of people in a crowd and their spatial locations by sampling from a posterior distribution over crowd configurations. Although this framework can be naturally extended from single to multiview detection, we show that the naive extension leads to an inefficient sampler that is easily trapped in local modes. We therefore develop a set of novel proposals that leverage multiview geometry to propose global moves that jump more efficiently between modes of the posterior distribution. We also develop a statistical model of crowd configurations that can handle dependencies among people and while not requiring discretization of their spatial locations. We quantitatively evaluate our algorithm on a publicly available benchmark dataset with different crowd densities and environmental conditions, and show that our approach outperforms other state-of-the-art methods for detecting and counting people in crowds.

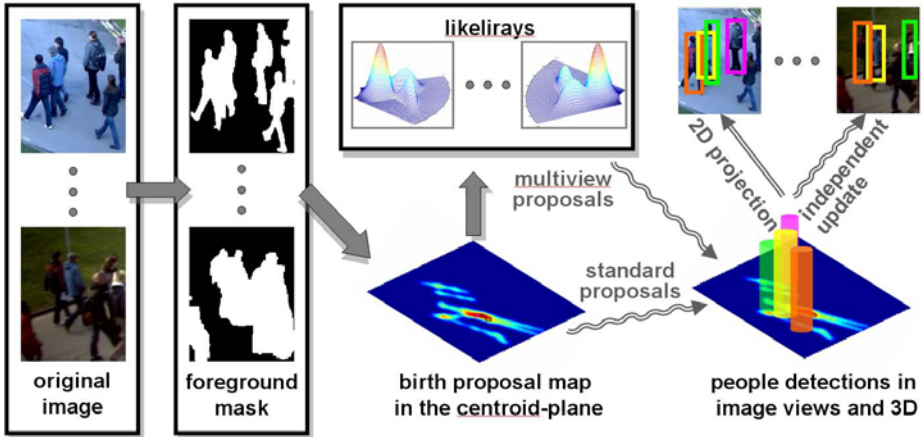
**Keywords:** Pedestrian detection; RJMCMC; Multiview geometry.

## 1 Introduction

Crowd detection is challenging due to scene clutter and occlusions among individuals. Despite advances in detecting and tracking people in crowds, monocular techniques are limited by ambiguities caused by insufficient information from a single view. Multiview approaches, on the other hand, can resolve ambiguities using complementary information from different views of the same scene. For example, two people totally overlapping in one view might be well separated in another view, making detection easier.

We present a probabilistic approach to estimate the *crowd configuration*, i.e. number of individuals in the scene and their spatial locations, regardless if people are visible in one view or multiple views. Our approach uses a stochastic process, specifically a Gibbs point process, to model the generation of multiview images of random crowd configurations. The optimal crowd configuration is estimated by sampling a posterior distribution to find the MAP estimate for which this generative model best fits the image observations. An overview of our approach is illustrated in Figure 1.

Our approach is motivated by the success of previous generative models for people detection [1,2,3]. Due to the great flexibility offered by sampling-based inference methods, our crowd model can accommodate inter-person dependencies that otherwise would be intractable to infer because of their inherent combinatorics. Efficient sampling strategies are the key to performance in practice.



**Fig. 1.** Our proposed method tests hypothesized crowd configurations in 3D space against multiview observations (foreground masks) within a sampling framework

Although various data-driven proposals have been designed in the single view context to guide hypothesis generation [2,3], to our knowledge we are the first to explore multiview geometry constraints for efficient sampling.

## Summary of Contributions

1. We extend generative sampling-based methods from single view to multiview, providing a unified framework for crowd analysis that successfully estimates 3D configurations in monocular and multiview input.
2. We introduce novel proposals based on multiview geometric constraints, yielding a sampler that can effectively explore a multi-modal posterior distribution to estimate 3D configurations despite occlusion and depth ambiguity.
3. Our global optimization does not require discretization of location and respects modeled spatial dependencies among people, resulting in better detection and localization accuracy than current state-of-the-art.

## 2 Related Work

Among monocular approaches for pedestrian detection [4,5,6,7,8,9], classifier-based methods are very popular [7,8,9] and sampling-based methods have also been shown effective for crowd detection [2,3,10] as well as generic object detection [11,12]. Within the sampling framework, various efficient, data-driven sampling strategies have been proposed. For example, Zhao and Nevatia [2] use a head detector to guide location estimates and Ge and Collins [3] learn sequence-specific shape templates to provide a better fit to foreground blobs. We extend the sampling framework to a unified approach that can detect people visible in a single view or in multiple views.

Previous multiview detection methods differ not only in image features and algorithms, but also camera layout. We confine our discussion to multiple cameras with overlapping viewpoints, for we are primarily interested in resolving ambiguities due to occlusion. Mittal and Davis [13] match color regions from all pairs of camera views to generate a ground plane occupancy map by kernel density estimation. In Khan et.al. [14], foreground likelihood maps from individual views are fused in a weighted average fashion based on field-of-view constraints. Tyagi et.al. [15] develop a kernel-based 3D tracker that constructs and clusters 3D point clouds to improve tracking performance.

Among related approaches that estimate ground plane occupancy [1,16,17,18], our work bears the closest resemblance to [1] in that we both take a generative approach. However, they discretize the ground plane into a grid of cells, and approximate the true joint occupancy probability of the grid as a product of marginal probabilities of individual cells, under the assumption that people move independently on the ground plane. Although our problem and framework are similar, we use a sampling-based inference technique that allows us to use a more flexible crowd model. Our model relaxes the independence assumption among people and does not require discretization of spatial location nor a fixed size for each person. We show in our results that these improvements lead to better detection and localization accuracy as well as greater robustness to errors in foreground estimation and camera calibration.

Our efficient sampling algorithm is inspired by previous work that seeks to improve the mixing rate of a sampler by encouraging traversal between different modes of the target distribution [19,20,21]. Dellaert et.al. [19] developed a chain flipping algorithm to generate samples of feasible solutions for weighted bipartite matching. Other methods such as the mode-hopping sampler [21] use knowledge about the topography of the target distribution to speed up sampling. Although inspiring, these methods are not directly applicable to our scenario because we are searching a large configuration space with variable dimension. More relevant is the data-driven MCMC framework [22] that uses various data-driven proposals such as edge detection and clustering to speed up Markov chain sampling for image segmentation.

### 3 A Gibbs Point Process for Crowd Detection

In this section we present a Bayesian statistical crowd model that accommodates inter-person dependence, together with a baseline sampling algorithm that directly extends a single view detection approach to perform multiview inference. We discuss the limitations of this baseline algorithm in Section 4 where we present the motivation and strategies of our novel multiview proposals. Experimental results on a public benchmark dataset are presented in Section 5.

#### 3.1 Modeling

Our goal is to estimate a 3D *crowd configuration* based on image observations from a surrounding set of fixed cameras. A crowd configuration is an unordered

set of targets  $\mathbf{o}^n = \{o_1, \dots, o_n\}$ ,  $i = 1, \dots, n$ ,  $n \geq 0$ . Each target represents a person moving on a flat ground plane and is parameterized by an upright cylinder  $o = (c, r, h)$ , where  $c \in W$  is a spatial coordinate in the centroid-plane, a plane that is half the height of an average person above the ground,  $W$  is a compact subset of  $\mathbb{R}^2$  equipped with volume measure  $\nu$ , and  $[r, h]$  specifies the width (radius) and height of a person.

The configuration space is denoted as  $\Omega_N = \{\emptyset, \cup_{i=1}^N \mathbf{o}^i\}$ , which is a union of subspaces with varying dimensions, including the empty set and up to  $N$  people distributed over  $W$ . We model random configurations by a spatial point process, specifically, the Gibbs point process [23]. Let  $\mu(\cdot)$  be the distribution of a homogenous Poisson process of unit intensity, which is analogous to the Lebesgue measure on  $\mathbb{R}^d$ . The density of the Gibbs point process can be defined with respect to this reference Poisson process. Formally,

$$p(\mathbf{o}) = \frac{f(\mathbf{o})}{\int_{\Omega} f(\mathbf{o}) d\mu(\mathbf{o})}, \quad (1)$$

where the mapping  $f(\mathbf{o}) : \Omega \rightarrow [0, \infty)$  is an unnormalized density having the Gibbs form  $f(\mathbf{o}) = \exp\{-U(\mathbf{o})\}$ .

The Gibbs process is very flexible for modeling prior knowledge about object configurations. It often includes a unary data term to model object attributes and higher-order interaction terms to model inter-object relationships. Our model incorporates two types of inter-person dependency. The first one is an avoidance strategy motivated by studies in social science showing that people keep a ‘comfort zone’ around themselves. We incorporate this dependency by a Strauss Model [23], which defines a pairwise potential interaction as

$$\phi(o_i, o_j) = \begin{cases} \eta & \|c_i - c_j\| \leq r \\ 0 & \|c_i - c_j\| > r \end{cases}, \quad (2)$$

where  $r$  is a parameter that controls the size of the comfort zone and  $\eta$  is set to some large constant number.

The second modeled dependency is based on the principle of non-accidental alignment. It penalizes configurations where people line up perfectly along a viewing ray to claim the same foreground region. This is not a hard constraint: certainly one person can be occluded by another in any view. However, each person is unlikely to be occluded in every view. In general, we seek to penalize configurations that require a large number of occlusions to explain the data. Unfortunately, explicit occlusion analysis involves a combinatorial number of interacting subsets. To keep the energy function linear in the number of targets, we measure the degree of alignment in 3D by the amount of overlap among projected rectangles in each image view. Formally, a ‘label’ image is computed by pixel-wise disjunction as  $\mathbf{S}^v(\mathbf{o}) = \cup_i \mathcal{H}^v(o_i)$ , where  $\mathcal{H}^v$  is projection function associated with camera  $v$  that maps a 3D person to a binary image that is zero everywhere except for a rectangular area bounding the projected person and  $v \in [1, V]$  where  $V$  is the number of camera views. Pixels in the label image covered by at least one projected rectangle are labeled as foreground. For simplicity, we

use  $\mathbf{S}^v$  as a shorthand for  $\mathbf{S}^v(\mathbf{o})$ . For each object  $o_i$ ,  $D_i^v = |\mathcal{H}^v(o_i) \cap \mathbf{S}^v(\mathbf{o} \setminus o_i)|$  measures the amount of overlap between one target’s projection and the rest of the targets in that view by counting the number of foreground pixels in the intersection image. We define the overlap cost for  $o_i$  as

$$D_i = \begin{cases} D_i^v & o_i \text{ only visible in } v \\ \min_v D_i^v & \text{otherwise} \end{cases} .$$

This way, overlap in some views will not be penalized as long as the target is clearly visible in other views. We encode prior knowledge about a general crowd configuration in the total energy of a Gibbs process

$$U(\mathbf{o}) = \sum_{i,j} \phi(o_i, o_j) + \sum_i D_i + \gamma N, \tag{3}$$

where  $N = |\mathbf{o}|$  is the number of estimated people in 3D. The last term penalizes spurious detections with a constant weight  $\gamma$ .

Under this probabilistic framework, the problem of crowd detection is solved by finding the configuration that best explains the image observations (foreground masks) from different views. Denote the binary foreground mask in view  $v$  by  $\mathbf{Z}^v = \{Z_i^v\}$ ,  $Z_i^v \in \{0, 1\}$ ,  $i = 1, \dots, m_v$ , where  $m_v$  is the number of pixels in the image observed from view  $v$ . A likelihood function  $\mathcal{L}$  is defined to measure the probability of a configuration given the foreground masks by comparing two sets of binary images, the mask images  $\mathbf{Z}$  and label images  $\mathbf{S}$ ,

$$\mathcal{L}(\mathbf{o}; \mathbf{Z}) = \mathcal{L}(\mathbf{S}; \mathbf{Z}) = \exp\{-G(\mathbf{o})\}, \tag{4}$$

$$G(\mathbf{o}) = \sum_{v=1}^V \sum_{i=1}^{m_v} I_1(S_i^v, Z_i^v) + \beta \sum_{j=1}^N I_2(o_j), \tag{5}$$

$$I_1(S_i^v, Z_i^v) = \begin{cases} 1 & S_i^v \neq Z_i^v \\ 0 & \text{o.w.} \end{cases}, \quad I_2(o_j) = \begin{cases} 1 & \exists v, \text{ s.t. } \frac{|\mathcal{H}^v(o_j) \cap \mathbf{Z}^v|}{|\mathcal{H}^v(o_j)|} < 0.1 \\ 0 & \text{o.w.} \end{cases}, \tag{6}$$

This likelihood function contains two terms:  $I_1$  penalizes discrepancies between hypothesized person detections and the image observations, and  $I_2$  imposes an extra penalty on ‘ghosts’ – detections that cover mostly background pixels.  $\beta$  is set to some large constant number.

Combining the prior (Eqn. 1) and the likelihood function (Eqn. 4), we define the optimal crowd configuration as the MAP estimator

$$\mathbf{o}^* = \arg \max_{\mathbf{o} \in \Omega} (P(\mathbf{o}|\mathbf{Z})) = \arg \max_{\mathbf{o} \in \Omega} \left( \frac{e^{-\left(U(\mathbf{o})+G(\mathbf{o})\right)}}{C(\Omega)} \right). \tag{7}$$

Optimizing the above posterior directly is intractable because the normalizing constant from the Gibbs prior,  $C(\Omega) = \int_{\Omega} f(\mathbf{o})d\mu(\mathbf{o})$ , involves all possible configurations in the combinatorial configuration space  $\Omega$ . Moreover, pairwise potentials in our crowd model make the inference harder than what can be handled by approximation methods such as [18,24].

### 3.2 Inference

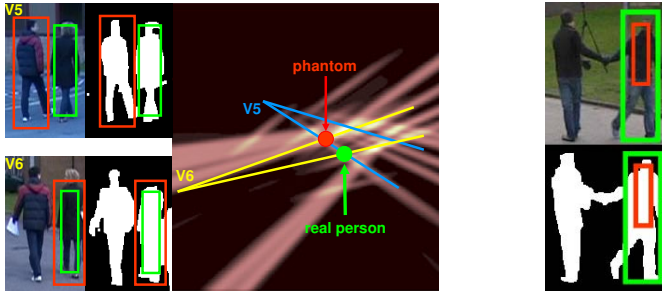
We use reversible jump Markov Chain Monte Carlo (RJMC) to battle the intractable normalizing constant in Eq. 7. MCMC is designed to generate samples from complicated target distributions, such as our posterior distribution, by constructing a Markov chain with the desired target distribution as its equilibrium distribution. RJMC [25] extends the classic algorithm to deal with variable dimension models. It suits the crowd analysis problem well because the number of people is not known a priori, and thus also needs to be estimated.

The RJMC sampler explores the configuration space by proposing perturbations to a current configuration. The general sampling framework is reviewed in the supplemental material [1]. The design of good proposal distributions is the most challenging part of the sampling algorithm. Proposals that only allow local perturbations may become trapped in local modes, leaving large portions of the solution space unexplored, whereas global adjustments have less chance to be accepted unless the target distribution is very smooth or tempered to be so. To achieve a good balance of both local and global proposals, we use proposals from a mixture of both types:  $Q(\cdot) = \sum_{c=1}^C p_c Q_c(\cdot)$ , where  $\sum_c p_c = 1$ ,  $\int Q_c(\mathbf{o}'; \mathbf{o}) \mu(d\mathbf{o}') = 1$ , and  $C$  is the number of different proposal moves. Below we describe a baseline multiview sampler directly extended from local birth, death, and update proposals commonly used in single view samplers [23].

**Birth/Death proposal.** A birth proposal adds a 3D person to the current configuration, i.e.  $\mathbf{o}' = \mathbf{o} \cup o_b$ . A simple birth strategy might place a person uniformly at random (u.a.r.) in the bounded region  $W$ . A death proposal removes a person from the current configuration so that  $\mathbf{o}' = \mathbf{o} \setminus o_d$ , e.g. choosing  $o_d$  u.a.r. from  $\mathbf{o}$ . Both proposals involve a dimension change from  $|\mathbf{o}|$  to  $|\mathbf{o}'|$ . Instead of blindly adding a person, we use a more informative data-driven proposal [22]. We sample  $o_b$ 's location according to the birth probability  $P_b \sim \frac{P_b(l)}{\sum_{l \in \bar{W}} P_b(l)}$ , where  $P_b(l) = \frac{1}{V} \sum_v \frac{|\mathcal{H}^v(l) \cap Z^v|}{|\mathcal{H}^v(l)|}$  is the fused occupancy likelihood of a particular location  $l$ , computed as the sum of the percentage of foreground pixels within its projected rectangles in all views, and  $\bar{W}$  is a discretization of the bounded region of interest in the centroid-plane  $W$ . Our final detection results are not restricted by this discretization because localization is adjusted by other proposals of the sampler.

**Update Proposal.** The update proposal preserves the dimension of the current configuration but perturbs its member's attributes (location and size) to generate a new configuration. We use a random walk proposal that selects a person  $o_u$  u.a.r. from  $\mathbf{o}$ , and either proposes a new spatial placement by sampling from a truncated normal distribution  $\mathcal{N}(c'|c_u, \sigma)$  centered at the current location  $c_u$ , or proposes a new size by sampling from a truncated normal centered at the size of an average person,  $h = 1.7m$  and  $r = 0.4m$ .

<sup>1</sup> <http://vision.cse.psu.edu/projects/multiviewmcmc/multiviewmcmc.html>



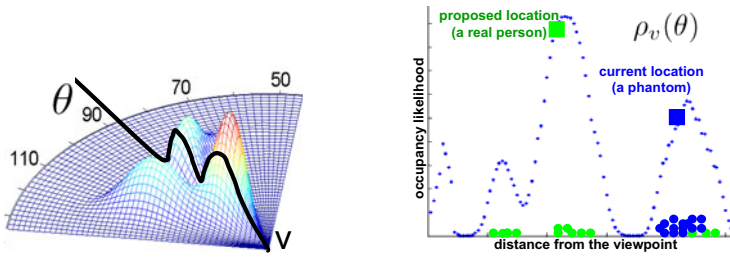
**Fig. 2.** Common pitfalls in multiview crowd detection. **Left:** the phantom phenomenon. A 3D phantom location (red circle) explains foreground pixels in different views that actually belong to the projections of two different real people (red boxes). **Right:** depth ambiguity for people visible in a single view can result in explanation of a single foreground region by alternative detections of different sizes.

## 4 Multiview Proposals

The local proposals presented in the previous section yield satisfactory results when people are well-separated in multiple views. However, when a person is visible only in one view, the inherent depth ambiguity coupled with noisy foreground blobs leads to a significant performance drop, which has also been reported in previous work [18,24]. Moreover, as occlusion becomes more frequent, we have observed that the naive sampler often gets stuck in local modes because of the ‘phantom’ phenomenon. Phantoms are accidental intersections of viewing rays at locations that are not occupied by any real person. Phantom hypotheses attempt to explain foreground regions across multiple views that actually are projections of different people in 3D. As shown in Figure 2, when a phantom gets accepted in the current configuration, later proposals for the real person are less likely to get accepted because the phantom already explains a large portion of their foreground pixels, thus the new birth proposal will suffer a high overlap penalty. Local random walk updates are also unlikely to escape from this local maximum. Although increasing the step size of a random walk can alleviate the problem to some extent, such blind exploration wastes time visiting mostly low probability regions, leading to an inefficient sampler.

Inspired by long range mode-hopping MCMC proposals [19,20,21], we exploit geometric constraints to design proposals that allow global changes that more effectively explore the configuration space. The motivation behind using geometric constraints is that multiview geometry is consistent across views whereas image-based appearance constraints (e.g. head detection for birth [2]) may conflict with each other in different views.

Our multiview proposals are based on occupancy likelihood rays, or *likeli-rays* for short. Recall that in our data-driven birth proposal, we have computed a centroid-plane occupancy map by fusing foreground masks from all the views in 3D. Likeli-rays are essentially polar coordinate transformations of the



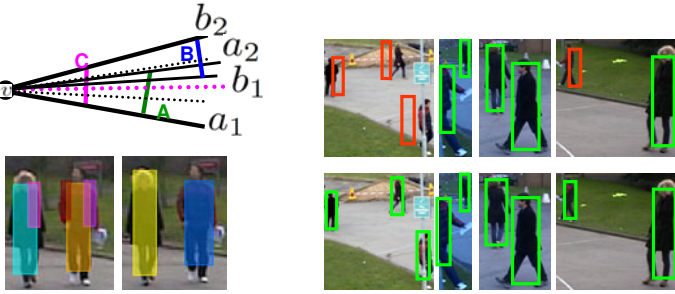
**Fig. 3.** **Left:** Likelihoods from one viewpoint  $v$ , indexed by angle  $\theta$ , define a set of 1D distributions over potential person and phantom locations at different depths along viewing rays in the centroid-plane. **Right:** Mode-hopping by sampling from a likelihood  $\rho_v(\theta)$ . Green dots are samples from the depth move, which visits all significant modes whereas the blue samples from a local random walk proposal stays in a single mode.

centroid-plane occupancy map with respect to each camera view  $v$ , indexed by angle  $\theta$ , i.e.  $\rho_v(\theta)$ . Different modes along each likelihood correspond to potential real and phantom locations of people at different depths. The likelihood representation gives us a convenient way to generate proposals with respect to a single camera view while taking into account fused information from all other camera views. We now present two such multiview proposals.

**Depth Move Proposal.** A depth move first randomly selects a person  $o_m$  and a camera view  $v$  from the list of views where  $o_m$  is visible. Let  $\theta$  denote the angle of the polar coordinate of  $o_m$ . A new 3D location is sampled with probability proportional to the 1D likelihood distribution  $\rho_v(\theta)$ . Figure 3 shows that samples from depth moves are able to visit different modes whereas samples from local random walk proposals only cluster around the current location. The depth proposal is a powerful and versatile mechanism to handle the problems shown in Figure 2. It can switch between a phantom and a real person hypothesis and also can produce the effect of a large scale change of a single person by “sliding” them in depth along a viewing ray, which is useful for correctly detecting people visible only in a single view. Unlike random walk with large step size, a depth move preserves some of the already covered foreground pixels. Depth moves therefore tend not to cause large decreases in likelihood, so are more likely to be accepted.

**Merge/Split Proposal.** When people are only visible in a single view and the viewpoint is not very elevated, a large foreground region may become covered by fragmented detections corresponding to pedestrian hypotheses scattered within a small range of viewing angles at different distances from the camera may be hypothesized to cover parts of a large foreground region (Figure 4). These fragments create local modes that prevent explaining the entire region correctly as one single person.





**Fig. 4. Left:** Merge Proposal. The top panel shows how the 3D merge proposal yields a new hypothesis  $C$  that minimally covers both projections  $A$  and  $B$  in view  $v$ . The bottom shows that the final results (right) correctly recover from fragmented hypotheses (left). **Right:** Independent Update Proposal. The top panel shows localization error (marked in red) in four views due to camera calibration and synchronization errors. The bottom shows improved results using the independent update proposal.

We design a 3D merge/split move to ease the switch between the following two hypotheses: multiple distant people versus a single, closer person. Let two people  $o_a$  and  $o_b$  both be visible from a particular viewpoint, with polar coordinates  $(\theta_a, r_a)$  and  $(\theta_b, r_b)$ ,  $\theta \in (0, \pi)$ . As illustrated in Figure 4, their angular extents are  $[a_1, a_2]$  and  $[b_1, b_2]$ . A new merged person  $o_c$  can be hypothesized from  $o_a$  and  $o_b$  in two ways: 1) when one of the angular extents completely falls within the other, we randomly move the person with the larger angular extent closer to the camera and delete the other; 2) otherwise, without loss of generality, assume  $a_1 < b_1$  and  $a_2 < b_2$ , which includes the case of partial occlusion as well as complete separation of the two. We create a new person in 3D whose image projection minimally covers the projections of both merge candidates, thus having an angular extent  $[a_1, b_2]$ . The corresponding polar coordinates  $(\theta_c, r_c)$  of  $o_c$  can be computed as  $\theta_c = \frac{a_1 + b_2}{2}$ ,  $r_c = \frac{0.5w}{\tan(0.5(b_2 - a_1))}$ , where  $w$  is the width of an average sized person.

A 3D merge move randomly chooses a view  $v$  in which to propose a merge. Denoting all visible people in  $v$  as  $\mathbf{o}_v$ , a person  $o_a$  is chosen u.a.r. from  $\mathbf{o}_v$ . For each other person  $o_i$ ,  $i \neq a$ , let  $e_i$  be the angular extent of the candidate blob that would result from merging  $o_a$  and  $o_i$ . We use these extents to define a probability distribution over candidates  $i$  as  $p_i = \frac{\tilde{e}_i}{\sum_j \tilde{e}_j}$ , where  $\tilde{e}_i = \frac{\min_j e_j}{e_i}$  favors merging two people with large angular overlap. A candidate person is proposed for merging with  $o_a$  by sampling from this distribution. If a newly merged person is accepted, we store their components in a merge list. The reverse proposal is a 3D split that randomly selects a person from the merge list and splits them back into their stored original component detections.

**Independent Update Proposal.** So far, the four types of presented proposals, birth/death, update, depth move, and merge/split, all hypothesize new person locations/sizes in 3D and the corresponding projections in image views are determined by the camera calibration information. To accommodate noisy input,

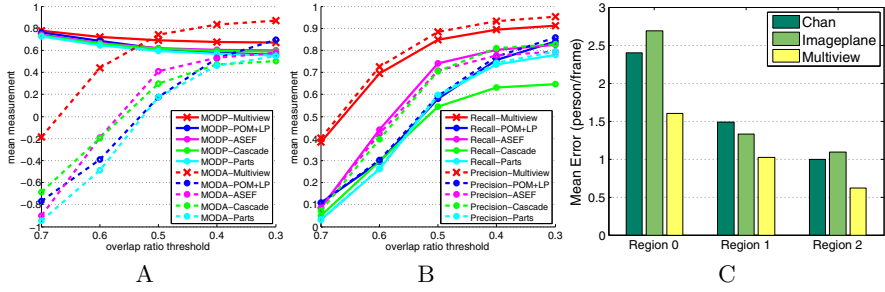
e.g. errors in calibration, synchronization, or foreground estimation, we add an independent update proposal that can perturb the 2D projection rectangle in each image plane independently (demonstrated in Figure 4). The independent update move works by randomly choosing a person  $o_i$  and a camera view  $v$  from the list of views where  $o_i$  is visible. With equal probability, either the size or the location of the projection box in view  $v$  is updated by sampling from a truncated 2D normal distribution centered at the nominal image location and size determined by the calibration matrices.

## 5 Experiments

We evaluate our algorithm on the PETS2009 dataset [26], a challenging benchmark dataset for multiview crowd image analysis containing outdoor sequences with varying crowd densities and activities. We tested on two tasks: crowd detection in a sparse crowd (sequence S2L1-1234) and crowd counting in a dense crowd (sequence S1L1-1357). We generated foreground masks using an adaptive background subtraction algorithm similar to Zivkovic’s method [27], and camera calibration information provided with each dataset was used to generate the birth proposal map  $P_b$  as the average back-projection of foreground masks from all views, as described in Section 3.2. Sample detection results are shown in Figure 6. Our proposed method obtains superior results over other state-of-the-art crowd detection methods, as will be shown through quantitative evaluation below.

**Sparse sequence S2L1:** We used four camera views, including one elevated, far field view (called View 1) and three low-elevation near field views with frequent, severe occlusions (Views 5, 6, and 8). We compared our detection results against the ASEF method, which is a detection method using convolution of learned average of synthetic exact filters [5], and the POM+LP method, which is a multi-target detection and tracking algorithm based on a probabilistic occupancy map and linear programming [24]. We chose these two methods because they are the current top-performers as reported in Winter-PETS2009 [26]. We also compared against the Cascade [8] and Part-based [9] person detectors, trained according to [5]. We performed ground-truth annotation of the sequence and evaluated each algorithm based on the standard MODA and MODP metrics (details are included in the supplemental material<sup>1</sup>). MODP measures localization quality of the correct detections and MODA measures detection accuracy taking into account false negatives/positives. For both metrics, larger values are better. A detection is counted as correct if the overlap ratio between the annotated box and the detection box is greater than some threshold  $\tau$ . We systematically vary this threshold and compute the evaluation metrics at each threshold. Correct detections and false positives/negatives are determined by solving an assignment problem between the annotations and the detection output.

Figure 5(A) shows averaged MODP and MODA scores across four views for our method and POM+LP, and over the detections from View 1 for the three classifier-based detectors (those are monocular methods that only have results



**Fig. 5.** Evaluation results on S2L1 and S1L1. For S2L1, our algorithm (red curves) consistently outperforms other methods in terms of MODA&MODP (A) and Precision&Recall metrics (B) at different overlap threshold levels without using temporal or appearance information. For S1L1 (C), we achieve lower count errors in all three target regions than current state-of-the-art methods.

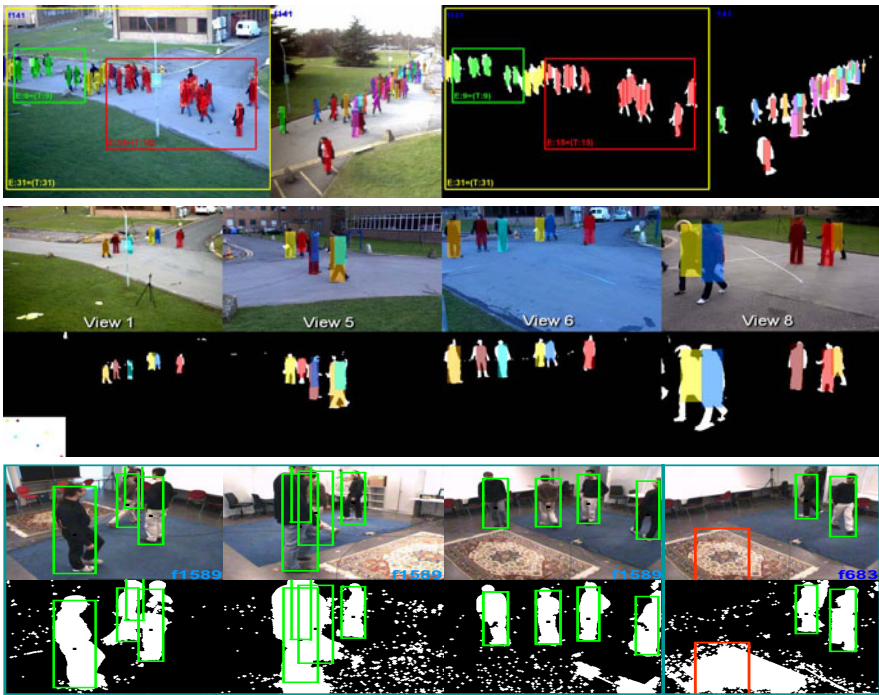
reported for View 1). Our multiview MCMC method consistently outperforms others (higher detection accuracy) at all overlap threshold levels. Additionally, the prominent performance gap at the tighter end of the threshold levels (larger  $\tau$ ) indicates that our method has better localization quality than other methods. It is interesting to note that our method is the top performer even though we do not use temporal consistency constraints across frames or discriminative object appearance information. Our improved accuracy is due to use of a more flexible generative model, made possible by the sampling-based inference, and our novel multiview proposals that allow more efficient global exploration of the posterior distribution. Since we are free from restrictions of discrete ground-plane grids, fixed 3D person size, and independence among people, we achieve better 3D localization than POM+LP, even with noisy input (Figure 6).

As the overlap threshold decreases, we see (as expected) an increase in MODA and decrease in MODP, since more misaligned detections become classified as correct. However, our MODP curve has a slower decreasing rate than others, which again confirms that we achieve better localization accuracy. Figure 5(B) shows results from a similar comparison but using precision/recall metrics. Our method has higher recall and precision than other methods.

In Table 1, we compare our multiview MCMC method to the naive baseline MCMC approach, introduced in Section 3.2, which does not use the new multiview proposals. The new multiview method outperforms the baseline approach in all cases. In the same table, we also show that our method works well with monocular sequences. For this experiment, we only use input observations from a single view. As opposed to the significant performance drop of the POM method reported in [24] in this situation, our single view detection results do not vary dramatically from the multiview results, and continue to outperform the other methods. These experiments indicate that our multiview proposals are effective at dealing with depth ambiguity, even along viewing rays from a single camera.

**Table 1.** MODP (1st column) and MODA (2nd column) in each view of S2L1 at an overlap threshold  $\tau$  of 0.5. Scores in bold indicate the top-ranked algorithm with respect to score metric and view. The first three rows are variants of our sampling-based approach and the bottom four are other state-of-the-art methods.

Method	View 1		View 5		View 6		View 8	
Multiview	0.6805	<b>0.7532</b>	<b>0.6872</b>	<b>0.6998</b>	<b>0.6953</b>	<b>0.8162</b>	<b>0.7004</b>	<b>0.6941</b>
Baseline	0.6791	0.6988	0.6872	0.5660	0.6936	0.6967	0.6967	0.5702
Singleview	<b>0.6863</b>	0.7052	0.6751	0.6415	0.6855	0.5333	0.6924	0.5357
POM+LP	0.5806	-0.1037	0.6071	0.2630	0.6467	0.3354	0.6344	0.2188
ASEF	0.6212	0.4116	-					
Cascade	0.6150	0.3000	-					
Parts	0.5927	0.1759	-					



**Fig. 6.** Sample detection results for S1L1 (top) and S2L1 (middle), overlaid on the original images and foreground masks. The bottom row shows sensitivity of our method to varying levels of noise in the foreground mask.

**Dense sequence S1L1:** The S1L1 sequence is captured from more elevated camera viewpoints, but with a higher crowd density and more lighting changes due to intermittent cloud cover. We annotated ground-truth person counts in all three regions specified by the PETS evaluation for View 1, shown in Figure 6.

and detect people using two camera views. The average count error for each region over the whole sequence is reported in Figure 5(C). Our error rate is less than 2 people per frame, better than the already remarkable results from Chan using holistic properties [28], which are the best results reported so far. We also compared against a 2D MCMC implementation [10] that performs birth, death and update proposals within the 2D image plane of View 1.

In Figure 6 we show sensitivity of this approach to errors in foreground estimation. This is an indoor sequence of 4 people walking [1]. On the left we see that our approach is tolerant of typical levels of foreground noise. However, as shown on the right, large areas of the image incorrectly labeled as foreground (due, for example, to failures of background subtraction to handle rapid lighting changes), can lead to false positive detections. However, our framework can be easily adapted to input data other than foreground masks, such as motion information or pedestrian classifier score maps.

## 6 Conclusion

We extend monocular, sampling-based crowd detection methods to perform multiview detection to accurately localize people in 3D given single or multiview images. Our results on a challenging benchmark dataset for crowd analysis demonstrate the advantage of our approach compared to other state-of-the-art methods. We have designed novel proposals that leverage multiview geometric constraints to effectively explore a combinatorial configuration space with varying dimension (numbers of people) while solving the problem of phantoms in multiview sequences and depth ambiguity in monocular sequences. Our sampling-based inference framework yields great flexibility in defining generative models that enable accurate localization of individuals in crowds despite occlusions, noisy foreground masks, and errors in camera calibration and synchronization.

**Acknowledgments.** We thank other PETS09 participants for sharing their results. This work was partially funded by NSF IIS-0729363.

## References

1. Fleuret, F., Lengagne, R., Fua, P.: Fixed point probability field for complex occlusion handling. In: ICCV (2005)
2. Zhao, T., Nevatia, R.: Bayesian human segmentation in crowded situations. In: CVPR (2003)
3. Ge, W., Collins, R.T.: Marked point processes for crowd counting. In: CVPR (2009)
4. Leibe, B., Seemann, E., Schiele, B.: Pedestrian detection in crowded scenes. In: CVPR, pp. 878–885 (2005)
5. Bolme, D.S., Draper, B.A., Beveridge, J.R.: Average of synthetic exact filters. In: CVPR, pp. 2105–2112 (2009)
6. Tu, P., Sebastian, T., Doretto, G., Krahnstoeber, N., Rittscher, J., Yu, T.: Unified crowd segmentation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 691–704. Springer, Heidelberg (2008)
7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: ICCV (2005)

8. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR, pp. 511–518 (2001)
9. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multi-scale, deformable part model. In: CVPR (2008)
10. Ge, W., Collins, R.T.: Evaluation of sampling-based pedestrian detection for crowd counting. In: Winter-PETS (2009)
11. Ortner, M., Descombes, X., Zerubia, J.: A marked point process of rectangles and segments for automatic analysis of digital elevation models. *TPAMI* 30, 105–119 (2008)
12. Rue, H., Hurn, M.: Bayesian object identification. *Biometrika* 86, 649–660 (1999)
13. Mittal, A., Davis, L.S.: M2Tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 18–33. Springer, Heidelberg (2002)
14. Khan, S.M., Shah, M.: Tracking multiple occluding people by localizing on multiple scene planes. *TPAMI* 31, 505–519 (2009)
15. Tyagi, A., Keck, M., Davis, J., Potamianos, G.: Kernel-Based 3D tracking. In: IEEE International Workshop on Visual Surveillance (2007)
16. Otsuka, K., Mukawa, N.: Multiview occlusion analysis for tracking densely populated objects based on 2-D visual angles. In: CVPR, vol. 1, pp. 90–97 (2004)
17. Yang, D.B., González-Baños, H.H., Guibas, L.J.: Counting people in crowds with a real-time network of simple image sensors. In: ICCV, pp. 122–129 (2003)
18. Alahi, A., Jacques, L., Boursier, Y., Vanderghynst, P.: Sparsity-driven people localization algorithm: Evaluation in crowded scenes environments. In: Winter-PETS (2009)
19. Dellaert, F., Seitz, S., Thorpe, C., Thrun, S.: EM, MCMC, and chain flipping for structure from motion with unknown correspondence. *Machine Learning* 50 (2003)
20. Andricioaei, I., Straub, J.E., Voter, A.F.: Smart darting Monte Carlo. *The Journal of Chemical Physics* 114, 6994–7000 (2001)
21. Sminchisescu, C., Welling, M., Hinton, G.: A Mode-Hopping MCMC Sampler. Technical Report CSRG-478, University of Toronto (2003)
22. Zhu, S., Zhang, R., Tu, Z.: Integrating bottom-up/top-down for object recognition by data driven Markov Chain Monte Carlo. In: CVPR, pp. 738–745 (2000)
23. van Lieshout, M.: Markov Point Processes and their Applications. Imperial College Press, London (2000)
24. Berclaz, J., Fleuret, F., Fua, P.: Multiple object tracking using flow linear programming. In: Winter-PETS (2009)
25. Green, P.: Reversible jump Markov chain Monte-Carlo computation and Bayesian model determination. *Biometrika* 82, 711–732 (1995)
26. Ellis, A., Shahrokni, A., Ferryman, J.M.: PETS 2009 and Winter-PETS 2009 results: A combined evaluation. In: Winter-PETS (2009)
27. Zivkovic, Z.: Improved adaptive gaussian mixture model for background subtraction. In: ICPR, vol. 2, pp. 28–31 (2004)
28. Chan, A., Morrow, M., Vasconcelos, N.: Analysis of crowded scenes using holistic properties. In: PETS (2009)

# A Unified Contour-Pixel Model for Figure-Ground Segmentation

Ben Packer<sup>1</sup>, Stephen Gould<sup>2</sup>, and Daphne Koller<sup>1</sup>

<sup>1</sup> Computer Science Department, Stanford University, CA, USA

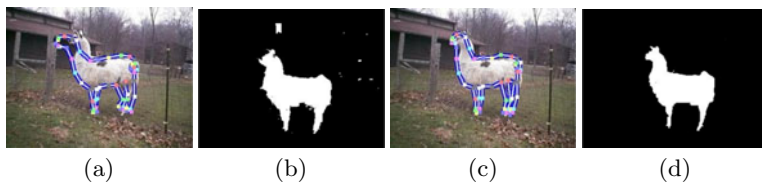
<sup>2</sup> Electric Engineering Department, Stanford University, CA, USA

**Abstract.** The goal of this paper is to provide an accurate pixel-level segmentation of a deformable foreground object in an image. We combine state-of-the-art local image segmentation techniques with a global object-specific contour model to form a coherent energy function over the outline of the object and the pixels inside it. The energy function includes terms from a variant of the TextonBoost method, which labels each pixel as either foreground or background. It also includes terms over landmark points from a LOOPS model [1], which combines global object shape with landmark-specific detectors. We allow the pixel-level segmentation and object outline to inform each other through energy potentials so that they form a coherent object segmentation with globally consistent shape and appearance. We introduce an inference method to optimize this energy that proposes moves within the complex energy space based on multiple initial oversegmentations of the entire image. We show that this method achieves state-of-the-art results in precisely segmenting articulated objects in cluttered natural scenes.

## 1 Introduction

The task of figure-ground segmentation is well established in the computer vision literature. There have generally been two types of approaches to this problem: outline-based methods (e.g., [2,3,4,5]) that denote the foreground by the interior of an object outline; and pixel-level foreground annotation (e.g., [6,7,8]) that label each pixel directly as either foreground or background. In this paper we combine these two approaches to achieve a superior and more refined object segmentation. Our method provides both an object contour, which exploits object-level information (such as shape), and a pixel annotation, which exploits pixel-level feature information (such as color and texture). We leverage this complementary relationship to improve the performance of each of these elements over using them in isolation.

We do so through two main contributions: The first, presented in Section 5, is the combination of the elements from two standard models for localization (contour) and segmentation into a unified energy model that can be precisely registered to a foreground object in a scene. Our model combines existing energy terms for each separate task ([1,4]) with an interaction term that encourages the contour and pixel-level segmentation to agree. Specifically, we introduce



**Fig. 1.** Contour and Segmentation. (a) Independent LOOPS outline. (b) Independent TextonBoost segmentation. (c) Joint model outline. (d) Joint model segmentation.

landmark-segment masks that capture the local shape of the foreground object in the vicinity of a single landmark or pair of landmarks along the object’s outline. Importantly, the masks are oriented and scaled to be consistent with the full object contour. This allows for a refined segmentation based on the articulated contour, which is not possible using a single global mask for the entire object. We also use the contour to construct an image-specific appearance model, which has been used successfully in other settings, further tying the two models. Example output for standard independent contour and segmentation models are shown in Figure 1(a) and (b), respectively. While each task produces reasonable initial results, our unified model leads to much improved figure-ground segmentation results, as shown in Figure 1(c) and (d).

Our second main contribution, presented in Section 6, is a method for optimizing the complex joint energy by proposing sets of moves within the entire search space, which is intractable to navigate in full. We build on the techniques of Gould et al. [9] by iteratively using the novel properties of our model to restrict the search space and efficiently finding a good solution within that subspace. Furthermore, this procedure lends itself to model-aware dynamic updates of the image-specific appearance model, which provides strong boosts in performance. In Section 7, we present experimental results to validate our approach, and show that we achieve both localized outlines and pixel-level segmentations that outperform state-of-the-art methods.

## 2 Related Work

Among successful object-specific, contour-based methods for object outlining are Ferrari et al. [5] (kAS) and Heitz et al. [1] (LOOPS). Our experimental results outperform both of these methods, and indeed we build on the latter to produce more accurate outlines. Among pixel annotation methods, the OBJ CUT method of Kumar et al. [7] and the method of Levin and Weiss [8] are two examples that, like our method, exploit both high-level shape cues and low-level features. They use these cues, however, in a strictly feed-forward manner to produce a segmentation. Our method propagates information both ways between the shape and pixel models, which results in a superior result for each one. Leibe et al. [6] do include a backprojection step that refines initial hypotheses. They do not, however, utilize a global model of object shape, nor do they produce a



single coherent result — their soft output allows cows to have more or less than four legs, for example.

Image-specific appearance models for object recognition have been used by Winn and Jojic [10], Kumar et al. [7], and Ramanan [11], among others. Our implementation learns this appearance with the help of a LOOPS model. This not only provides a particularly strong cue for using the correct pixels, but also allows us to use the properties of LOOPS to select those pixels carefully. As we describe in Section 4, we use the contour model to rate our uncertainty over different locations in the image, which allows us to learn the appearance only over pixels about which we are confident.

Our work is most similar to Bray et al. [12] and Chen et al. [13], which both combine a CRF-based segmentation model with an object model, as we do. The differences between our approach and theirs highlight our contributions. Bray et al. [12] use a single distance function to relate the object skeleton to the background segmentation. This is roughly equivalent to using masks as we do, but in their case these masks are the same for each part of the object and are restricted to the form of a distance function that does not capture outline detail. Chen et al. [13] use a single mask for the entire object, which is problematic for articulated objects since it cannot account for multiple configurations. Indeed, they report results on classes from Caltech 101 [14] that have rigid shapes and for which segmentation is easier than in cluttered scenes. In contrast, our landmark-specific masks are different for each part of the object, have a general form that can capture outline detail, and are learned from data to capture this detail. This allows us to learn and preserve particular shapes in the segmentation over different object parts such as the outline (and ears) of the head, even in the presence of articulated skeletons such as those found in the Mammals dataset [15], for which we report results. Furthermore, both [12] and [13] alternate between optimizing over the object and segmentation using coordinate ascent. We present an efficient method for joint inference, which can avoid local minima found in each task separately.

### 3 Localization and Segmentation Models

Our aim is to build a model that encompasses both the localization and the segmentation task, and that incorporates the interactions between the two in order to improve performance on each task. This model is specified by an energy function  $\Psi$  that is an aggregation of individual energy terms over various components of the model. In this section, we describe two approaches from the vision literature for solving the two separate tasks, each of which yields individual energy terms. We describe how these tasks can be solved separately as baseline methods, and in later sections we use these energy terms in our joint model. In Section 4, we introduce an interaction from the localization component to the segmentation component through image-specific features. In Section 5, we introduce landmark-segmentation masks that tie the two main model components together in a bidirectional manner.

### 3.1 Outline Localization

The recent LOOPS model of Heitz et al. [11] treats object localization as a landmark correspondence problem, the solution to which defines a piecewise-linear contour around the object in an image. We describe this model throughout this section. Formally, the task is to assign each landmark  $L_i$  to the appropriate pixel on the object’s outline. We denote the full assignment to all landmarks by  $\mathbf{L}$ .

Registering the landmarks to a test image requires optimizing an energy function  $\Psi^L(\mathbf{L})$  over the landmark assignments. This energy function is composed of two types of terms over the landmark assignments. The first is a singleton feature-based term that predicts the location of a specific landmark from a set of image features. We let  $\psi_i^L = \langle \theta_i^L, \phi_i^L \rangle$ , where  $\phi_i^L$  is the response vector of a boosted detector [16] for landmark  $i$ , and  $\langle \cdot, \cdot \rangle$  denotes the dot-product between the model parameters  $\theta_i^L$  and the landmark features  $\phi_i^L$ .

The second term in  $\Psi^L$  is a global shape term that gives preference to the landmarks forming a likely object shape. This term is a multivariate Gaussian over all landmarks, which decomposes into pairwise terms:

$$\delta_{i,j}^L = -\frac{1}{2}(L_i - \mu_i)\Sigma_{ij}^{-1}(L_j - \mu_j), \quad (1)$$

where  $\mu_i$  is the mean location of landmark  $i$  and  $\Sigma$  is the covariance matrix that relates the positions of all landmarks.

Figure 1(a) shows an example result of finding the optimal assignment over the landmark variables of the entire landmark energy:

$$\Psi^L(\mathbf{L}) = w_1 \sum_i \psi_i^L + w_2 \sum_{i,j} \delta_{i,j}^L, \quad (2)$$

where the weights  $w_1$  and  $w_2$  determine the relative influence of each term. The parameters and weights can all be learned from supervised data, and the energy can be optimized approximately in isolation using max-product message passing algorithms (see Section 6).

### 3.2 Foreground Segmentation

We now turn to a standard technique for foreground-background segmentation. This task amounts to assigning a variable  $S_k$  for each image pixel  $k$  to be either foreground ( $S_k = 1$ ) or background ( $S_k = 0$ ). The full assignment to all pixels is denoted by  $\mathbf{S}$ . We use a variant of the TextonBoost algorithm [17] to perform this task. Since the datasets we consider in Section 7 generally consist of a single foreground object on a background that is comprised of several common categories (such as grass, sky, and trees), we train a separate binary boosted classifier for each of these classes. The outputs of these classifiers are used as features for a logistic classifier that predicts whether each pixel is foreground. We use a pairwise binary conditional Markov random field (CRF) over the pixels in the image, where the singleton potentials are represented by the logistic

classifier and the pairwise potentials encourage neighboring pixels with a similar appearance to have the same label.

The CRF for foreground segmentation represents an energy  $\Psi^S(\mathbf{S})$  over the pixel assignments that consists of a singleton term  $\psi_k^S$  and a pairwise term  $\delta_{k,l}^S$ . Given the outputs of the various boosted classifiers for each pixel  $k$  in feature vectors  $\phi_k^S$ , the first term takes the form  $\psi_k^S = \langle \theta_{s_k}^S, \phi_k^S \rangle$ , where  $\theta_{s_k}^S$  is the set of logistic regression weights (shared between all pixels) associated with the assignment  $S_k = s_k$ . The pairwise term takes the form

$$\delta_{k,l}^S = \begin{cases} \exp\left(-\frac{\|c_k - c_l\|_2^2}{2\bar{c}}\right), & \text{for } (k,l) \in \mathcal{N}(\mathcal{I}) \text{ and } S_k \neq S_l \\ 0 & \text{otherwise} \end{cases}, \quad (3)$$

where  $\mathcal{N}(\mathcal{I})$  is the set of neighboring pixels in image  $\mathcal{I}$  (in our implementation we use 4-connected neighbors),  $c_k$  is the vector of *Lab* color values at pixel  $k$ ,  $\|\cdot\|_2^2$  is the  $L_2$  distance between such vectors, and  $\bar{c}$  is the mean such distance across all neighboring pixels in the image. Note that the pairwise term is only non-zero when neighboring labels that are not equal (i.e., at the boundary between foreground and background), and thus penalizes neighboring pixels when their labels are different and the contrast between them is low. The full segmentation energy over  $\mathbf{S}$  is given by

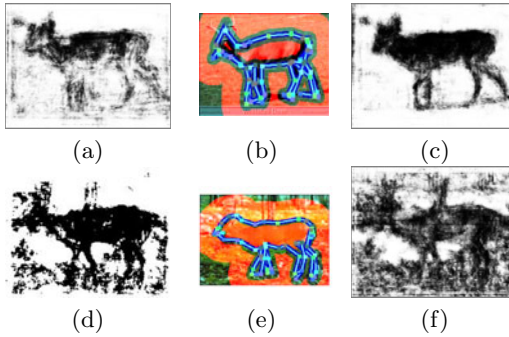
$$\Psi^S(\mathbf{S}) = w_3 \sum_k \psi_k^S + w_4 \sum_{k,l} \delta_{k,l}^S, \quad (4)$$

where  $w_3$  and  $w_4$  weight the two terms. As with the landmark model, the classifiers and weights are learned from the labeled training set. The energy can be optimized exactly in isolation using a graph cut [18] (see Section 6). Figure 11(b) shows an example result for the image in Figure 11(a).

## 4 Image-Specific Appearance

Building an *image-specific* appearance model helps combat the fact that the variation across images in the appearance of both the object class and background make it difficult or impossible to reliably separate the two. While the segmentation CRF models the fact that an object should have consistent appearance (at least in neighboring pixels) through its pairwise terms, the singleton terms nevertheless adhere to a single appearance model across the entire object class. We therefore use the initial localized outline of the LOOPS model to construct an *image-specific* appearance model to augment the class-level appearance model within the segmentation CRF at test time.

Specifically, we build a naive Bayes classifier based on pixel color values that will distinguish between the object in the image and the background particular to the image. To estimate the parameters of the classifier, we split the image pixels, each of which carries a class label of either foreground or background based on the contour estimate, into three mutually-exclusive sets: **E** (excluded), **C** (certain pixels), and **U** (uncertain pixels). Background pixels that are far



**Fig. 2.** Local appearance features. (a,d) Response maps of class-level boosted classifiers for deer. (b,e) Initial LOOPS outlines. Highlighted pixels are those chosen as “confident” training examples for the local appearance. (c,f) Response maps of the resulting appearance model.

away from the border of the localized contour are neither useful for training nor important to consider relabeling, and hence belong in **E**. Certain pixels, **C**, are non-excluded pixels (either foreground or background) for which the contour model is sufficiently confident about their label (see below). The remaining pixels belong in **U**. We train the naive Bayes model over only the pixels in **C** and **U** as follows: (1) we seed the class labels for the pixels in **C** based on whether the pixel is inside or outside the contour, (2) leave the class labels for **U** hidden, and (3) use the EM algorithm [19] both to learn an appearance model for the foreground and background, and to reinfer the class labels. The log of the posterior probability of each pixel being the foreground is then used as a feature — alongside the boosted classifier outputs (see Section 3) — for the logistic classifier component of the segmentation CRF, which is retrained.

To determine which pixels belong in **C**, we note that we may be more confident about certain parts of the object than others; for example, the localization method may be certain that it has localized the torso of the deer, but less certain about the particular placement of the legs. We determine the reliability of each landmark separately by measuring how likely the localization method is to have properly assigned that landmark. Let  $\sigma_i$  be the standard deviation of the distance of the localized landmark  $L_i$  to the true outline on the training data. We compute a signed distance  $Dist(k)$  (also used in Section 5) of each pixel in the test image to the localized outline, where the sign is positive if the pixel is inside the contour and negative otherwise. Pixel  $k$  belongs to **C** if  $|Dist(k)| > \sigma_i$  for the closest landmark  $i$ . Note that computing this score, as well as retraining the CRF’s logistic classifier, requires running the localization method on the training data. Figure 2 shows the responses of this naive Bayes classifier on a test image. In the top row, despite the imperfect LOOPS outline, the learned appearance model is still strong. However, as shown in the bottom row, even with a good LOOPS outline, the local appearance is not always a perfect feature. In Section 7, we analyze the results of augmenting the segmentation task in this way, which we refer to as **ImgSpec**.

## 5 The Contour-Pixel Model

We now present a unification of the contour and pixel models in which we incorporate more information than pixel appearance. Importantly, this information flows both ways. There is a natural agreement between localization and segmentation in that the pixels inside the contour outlined by the landmarks  $\mathbf{L}$  should be labeled as foreground, and those outside should be labeled as background.

A naive way to combine the two signals is simply to merge the segmentation CRF’s probability over each  $S_k$  with that pixel’s signed distance to the localized contour. Let  $P_0(S_k)$  be the posterior probability over  $S_k$  according to the CRF. We define our new probability  $P_1(S_k)$  to be the product of  $P_0(S_k)$  and the sigmoid of the signed distance  $Dist(k)$  (defined in Section 4), normalized to sum to one.

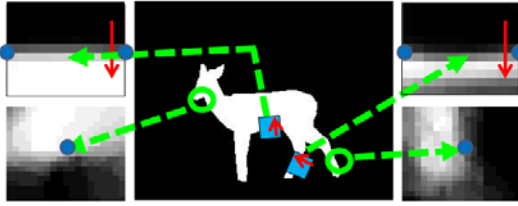
As we show in Section 7, combining the models in this way (which we call **Product**) does not lead to an improvement in performance. To fully exploit these parallel signals, rather than post-processing their outputs, we would prefer to allow each method to reoptimize its own variables in light of information propagated from the other. We now describe a model that unifies the two tasks in a single coherent model.

We introduce a new energy term  $\Psi^{L,S}(\mathbf{L}, \mathbf{S})$  that encourages agreement between landmarks  $\mathbf{L}$  and segmentation  $\mathbf{S}$ . Since a LOOPS landmark is a consistently located element of the object’s shape, the nearby pixel annotations should follow a pattern particular to that part of the object. For example, the pixels above the landmark corresponding to the stomach will generally belong to the foreground, while those below it will generally be part of the background. For each landmark  $L_i$ , we build an “annotation mask”  $M_i$  of size  $N_1 \times N_1$  that is a grid whose  $(a, b)$ -th entry indicates the probability that a pixel offset by  $(a, b)$  from the location of  $L_i$  is a foreground pixel. Each mask is learned from training images by aggregating masks of size  $N_1$  around the groundtruth landmark location in each training image, and the learned mask is simply the average of each of these masks. Examples of landmark masks near the nose and leg of a deer are shown in Figure 3. The energy term associated with this pairwise mask is

$$\psi_{i,k}^{LS_1} = S_k \log M_i(a, b) + (1 - S_k) \log(1 - M_i(a, b)). \quad (5)$$

If the offset  $(a, b)$  between landmark  $i$  and pixel  $k$  extends beyond the size of the mask ( $\frac{N_1}{2}$ ), then there is no pairwise energy term that relates  $L_i$  to  $S_k$ . This potential allows information to propagate between the contour model and the segmentation model. A landmark  $L_i$  with a high probability of appearing at a given location will encourage the surrounding pixels to be annotated according to the mask. This information can then propagate to the rest of the image pixels via  $\Psi^S$ . Conversely, a pattern fitting the mask appearing in the pixel labels encourages the landmark to assign itself in the appropriate nearby location, and this can influence the rest of the landmarks via  $\Psi^L$ .

In addition to masks that capture the relationship between single landmarks and their surrounding pixels, we introduce masks  $M_{i,j}^l$  that tie neighboring pairs



**Fig. 3.** Landmark-segment masks. The green arrows indicate the mask associated with various landmarks, which are marked as blue dots. The upper two masks are pairwise masks between neighboring landmarks, and are reoriented and rescaled appropriately — the red arrow indicates the “inside” direction of the mask.

of landmarks  $L_i$  and  $L_j$  jointly to their surrounding pixels. These masks are similar to  $M$ , but account for different orientations of consecutive landmarks. Each adjacent pair of landmarks  $L_i, L_j$  is associated with an oriented and scaled mask  $M'_{i,j}$  whose  $(a, b)$ -th entry is the foreground probability of the pixel offset by  $(c, d)$  from the midpoint between  $L_i$  and  $L_j$ , where  $(c, d)$  is found by rotating the vector  $(a, b)$  by the angle of the segment  $L_i - L_j$  and dividing by the length of that segment. We learn these pairwise masks from training data similar to the singleton masks above. The energy term associated with this mask is

$$\psi_{i,j,k}^{LS_2} = S_k \log M'_{i,j}(a, b) + (1 - S_k) \log(1 - M'_{i,j}(a, b)). \quad (6)$$

Figure 3 shows an example of a such a mask for consecutive landmarks along one of the deer’s hind legs. It clearly indicates that, regardless of the orientation of the leg, pixels that are on the “inside” of the line segment on the neck are more likely to be foreground.

Now that we have created the energy terms that tie together the variables of our model, we define the energy of a full variable assignment  $(\mathbf{S}, \mathbf{L})$  given the image as

$$\Psi = \sum_t w_t \cdot \Psi^t(\mathbf{S}, \mathbf{L}), \quad (7)$$

where  $t$  ranges over the types of energy terms. While weight ratios learned for each model are kept fixed, the relative weights for all terms are learned using cross-validation on the training set. Note that  $\Psi$  is composed of at most triple-wise terms between the variables  $\mathbf{S}$  and  $\mathbf{L}$ . Having defined this CRF over  $\mathbf{S}$ ,  $\mathbf{L}$ , and input image  $\mathcal{I}$ , we seek the single joint assignment to  $\mathbf{S}$  and  $\mathbf{L}$  that minimizes the energy. That is, the MAP solution is  $(\mathbf{S}^*, \mathbf{L}^*) = \operatorname{argmin}_{\mathbf{S}, \mathbf{L}} \sum_t w_t \cdot \Psi^t(\mathbf{S}, \mathbf{L})$ .

## 6 Superpixel-Based Inference

### 6.1 Inference Challenges

We now consider the properties of our coherent energy function in deciding how to optimize it. The pixel annotation terms  $(\Psi^S)$  can be optimized exactly using

a graph cut [18] if considered independently, since there are regular pairwise terms between binary-valued variables. However, the landmark location terms ( $\Psi^L$ ) cannot be optimized exactly even if considered independently, and in fact performing inference with these terms proves to be a challenge. To complicate matters, we have pairwise terms between pixels and landmarks (which can take many values) and triplewise terms between pixels and pairs of landmarks. A model with 50 landmarks in a  $300 \times 200$  pixel image, for example, would have 3 million pairwise terms and 150 million triplewise terms. Thus, there is a great deal of interconnectivity between the variables, and even constructing a graph to represent the full joint energy may be intractable.

### Coordinate Descent Baseline

One straightforward approach to inference would be to simply perform coordinate descent on the full energy. This can be done by first optimizing  $\Psi^L$  over  $\mathbf{L}$ , then folding the potentials in  $\Psi^{L,S}$  evaluated at the fixed  $\mathbf{L}$  into the singleton terms  $\psi_k^S$ , then optimizing  $\Psi^S$  separately over  $\mathbf{S}$  (which, again, may be done exactly and efficiently), then folding  $\Psi^{L,S}$  evaluated at the fixed  $\mathbf{S}$  into the singleton potentials of  $\Psi^L$ , and iterating back and forth in this manner. As we show in Section 7 this approach (which we call **Coord**) succeeds in sharing the signals between the two energies, but is susceptible to local minima and does not allow the exploration of the full variable space.

## 6.2 Joint Inference

To overcome these inference issues, we develop a search strategy for dealing with MAP inference in the face of such a complex and large search space by exploring dynamically constructed discrete subspaces. We then use a final refined stage, initialized from the result of the discrete stage, that uses the full search space.

Our joint inference algorithm proceeds as follows. We begin with an initial assignment to all of the variables, and then find a naturally defined and much smaller subspace through which we can explore the energy function. This subspace is defined by a set of proposal moves from the current assignment to new assignments to the variables. After performing inference within the simpler subspace, if the new assignment achieves an improved energy (note that since inference is not exact, we cannot guarantee that we have found the optimal assignment within the subspace), we keep the new assignment, and otherwise revert to the previous assignment. We then construct a new subspace and repeat.

### Constructing Search Subspaces

We choose a subspace for each iteration in two ways. The first stems from the observation that groups of nearby pixels tend to have the same label, and the relationship between landmarks and nearby pixels tends to be the same for entire groups of pixels. We therefore divide the image into superpixels (using the mean-shift segmentation algorithm [20]) and define our proposal moves over superpixel regions. Specifically, given a starting assignment, the proposal moves assign all pixels within a superpixel to either background, foreground, or their current assignment. This approach is similar to the search strategy proposed by Gould

et al. [9]. The inference problem can thus be recast in terms of region variables  $\mathbf{R}$  that can take on one of three values rather than individual pixel variables  $\mathbf{S}$  that can take on two values (foreground or background).

To avoid committing to any single oversegmentation of image, we use a different oversegmentation (by varying the parameters of the mean-shift algorithm) in each iteration. Each oversegmentation proposes a different set of moves within the space of pixel assignments. For example, an oversegmentation with a small number of large superpixels might propose assigning every pixel in the torso of the deer to the foreground, while a finer-grained oversegmentation might propose refining the pixel assignments around the edge of the torso.

The second simplification of the search space is a restriction on the values of the landmarks, and corresponds to the pruning proposed in Heitz et al. [11]: Rather than consider all pixels as possible assignments for the contour landmarks, we choose a small subset (of size  $K = 25$ ) of likely pixels as candidates in each round. Performing multiple rounds of inference, however, allows for the flexibility of choosing candidates for each round in a more dynamic and sophisticated way. In each round, we choose the landmark candidates to be the most likely pixels according to the singleton feature energy terms, subject to two restrictions that vary by round. First, we require that the candidates lie on a superpixel border (recall that the oversegmentations change each round). Second, we restrict each landmark to fall within two standard deviations of its mean location *given* the location of all other landmarks from the previous round of inference. Since the joint model over all landmarks is a Gaussian, computing the conditional Gaussian is straightforward. This restriction allows us to take advantage of the *global* shape information as well as cues from previous rounds. By restricting the search space in these two ways, for a 50-landmark model in a  $300 \times 200$  image that is split into 300 superpixels, the landmark search space is reduced from  $50^{60,000}$  to  $50^K$  and the segmentation task is reduced from a binary problem over 60,000 variables to a ternary problem over 300 variables.

### Inference Over Multiple Subspaces

Note that, although we construct a different inference model in each round, the algorithm always optimizes a single, consistent energy function. What differs in each round is the way in which the energy terms are combined and the set of moves that may be taken.

Once we have constructed the simplified inference model over the search subspace, we use residual belief propagation (RBP) [21] to perform MAP inference. The ability to do so efficiently depends on the important property of  $\Psi$  that it is composed of at most triple-wise terms between the component variables (the regions  $\mathbf{R}$  and the landmarks  $\mathbf{L}$ ). Specifically, the decomposition of  $\Psi^L$  presented in Section 3 uses only singleton and pairwise terms between the landmarks  $\mathbf{L}$ , and similarly the decomposition of  $\Psi^S$  uses only singleton and pairwise terms between the pixel labels  $\mathbf{S}$ , which translates into the same property over the smaller set of region labels  $\mathbf{R}$ . Finally, the landmark-pixel masks  $M$  result in pairwise terms between a single landmark  $L_i$  and a single region  $R_k$ , and the oriented masks  $M'$  result in triplewise terms between a pair of neighboring



landmarks and a single region. Consequently, RBP is able to converge quickly to a joint solution over all variables  $\mathbf{L}$  and  $\mathbf{R}$ . We experimented with other inference algorithms, such as dual decomposition [22], which generally achieved the same energy solutions as RBP.

### Final Refined Stage

Once this iterative process has converged, we reintroduce the full landmark domain and perform a final refined inference step as in LOOPS, allowing the contour landmarks to lie anywhere in the image. As a post-processing step, since our model defines a closed contour over the foreground object, we set all pixels outside the contour (with a buffer of size  $\delta = 5$  pixels) to be background. Though this post-processing step operates outside of the framework of the unified energy, it is not a deficiency of the energy construction itself. It is necessary to set pixels that are beyond the reach of the landmark masks to be part of the background. In principle, if the mask sizes were large enough, this step would not be necessary. However, the mask sizes must be kept reasonably small to avoid an overly dense connectivity among the variables. As a result, there is no term in the energy to discourage these faraway pixels from being set to the foreground.

## 7 Experimental Results

To validate our approach, we ran our method on several classes from the Mammals [15] and Caltech [14] datasets. For each class, we average over five random folds of the data with 20 images for training and the remaining (20-50) for testing. We obtained groundtruth segmentation labels using Amazon’s Mechanical Turk to augment existing contour labels for these datasets.

Because our task involves both locating the object landmark points and the annotated foreground-background segmentation, we present several metrics to evaluate the success of our method. The first is the simple pixel accuracy of the segmentation (percent of total pixels accurately labeled as foreground or background compared to the groundtruth segmentation). The second measures the accuracy of the precise contour implied by the annotated segmentation. We take the gradient of both the assigned segmentation and the groundtruth segmentation, dilate each by 5 pixels, and then compute the Jaccard similarity (intersection divided by union) between the two. The third metric is the symmetric outline-to-outline root-mean-squared (RMS) distance between the outline created by the assigned landmarks and the groundtruth outline.

The first baseline for comparison with our model is the **Independent** model that separately considers the landmark points and the annotated segmentation. That is, this baseline uses the implementations of TextonBoost and LOOPS in isolation as described in Section 3, utilizing neither the image-specific appearance features nor the landmark-segmentation masks in  $\Psi^{L,S}$ . We are thus comparing to a standard method for segmentation as well as a state-of-the-art method for landmark localization. For the **ImgSpec** baseline, specified in Section 5, the contour is used to learn the image-specific appearance, and the probability over  $\mathbf{S}$  according to the segmentation model is simply multiplied by a similar probability

**Table 1.** Outlining and Segmentation Results. Best performance is in bold; multiple results are in bold when the differences are statistically insignificant.

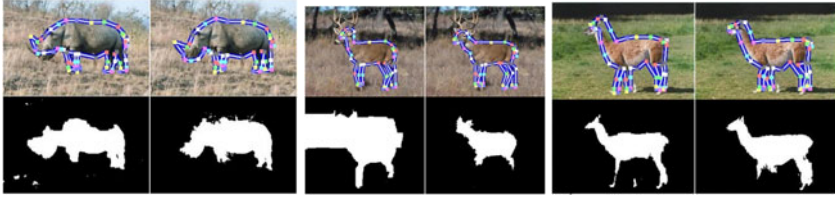
	Pixel Accuracy			Jaccard Similarity			RMS Distance	
	<b>Indep</b>	<b>ImgSpec</b>	<b>Joint</b>	<b>Indep</b>	<b>ImgSpec</b>	<b>Joint</b>	<b>Indep</b>	<b>Joint</b>
bison	<b>96.6</b>	<b>96.3</b>	<b>96.4</b>	78.6	79.0	<b>81.2</b>	4.0	<b>3.9</b>
elephant	90.5	92.2	<b>93.3</b>	71.7	70.8	<b>76.1</b>	<b>4.7</b>	<b>4.7</b>
llama	89.7	89.4	<b>93.0</b>	61.8	64.1	<b>73.4</b>	6.4	<b>5.3</b>
rhino	91.0	94.0	<b>95.1</b>	64.5	73.3	<b>75.7</b>	4.7	<b>4.4</b>
deer	88.7	89.5	<b>92.1</b>	56.9	54.8	<b>61.6</b>	8.9	<b>7.0</b>
giraffe	89.9	92.0	<b>92.6</b>	62.0	64.9	<b>65.8</b>	<b>6.4</b>	6.7
airplane	92.3	<b>96.3</b>	<b>96.6</b>	60.8	<b>74.7</b>	<b>74.6</b>	4.2	<b>4.0</b>
bass	92.5	92.5	<b>93.5</b>	58.4	<b>60.1</b>	<b>60.5</b>	10.7	<b>9.5</b>
buddha	84.4	86.0	<b>91.9</b>	42.2	44.7	<b>56.8</b>	10.8	<b>10.6</b>
rooster	91.3	92.1	<b>95.5</b>	57.9	61.1	<b>63.6</b>	10.8	<b>9.4</b>

according to the landmark contour. We refer to our method of optimizing the full energy  $\Psi$  jointly over all variables, as well as using the image-specific appearance, as **Joint** in the results that follow. Though we do not show the results here, the **Product** baseline from Section 5 did not outperform **Independent**.

The results for the classes considered are presented in Table 1. Our **Joint** method achieves a marked improvement over the **Independent** methods. It achieves higher pixel accuracy than the baseline segmentation on all classes except “bison,” for which the accuracy is statistically the same. All other differences are statistically significant according to a paired t-test: the least significant difference was the “bass” class with a mean difference of 1.0% and p-value of 0.003. For the outline similarity metric, our method was better on all classes, with the least significant difference being the “bison” class with a mean difference of 2.6% and a p-value of  $10^{-6}$ . For the landmark-based RMS distance, our model is statistically similar to the independent LOOPS on the “elephant” class, worse on “giraffe,” and better on all other classes despite small differences for some of them. The mean difference for the “bison” class is 0.1 pixels, but the paired t-test yields a p-value of 0.028. All other classes had significant differences, with the least significant p-value being  $10^{-5}$ .

Note that simply using the image-specific features (**ImgSpec**) gives a boost in segmentation over the baseline, but does not achieve the same level of results as using our full energy and inference. The full model’s pixel accuracy is superior on 7 out of the 10 classes, with all differences being statistically significant, while there is no statistical difference between the other 3 classes. For the outline similarity metric, the full model is superior on all classes except for “airplane,” for which the instances have relatively uniform appearance so that our outline-aided image-specific features account for all of the improvement in our method.

We also compared to the **OBJ CUT** method of Kumar et al. [7] and the **kAS Detector** method of Ferrari et al. [5], using downloaded code to run on these datasets. On the pixel accuracy, outline Jaccard similarity, and outline RMS scores, our **Joint** model outperforms the **kAS Detector** by macro-averages



**Fig. 4.** Three representative model segmentations. Each panel in the left column is produced separately by the **Independent** methods and the right column is produced by our **Joint** method.

over all classes of 3.2%, 1.5%, and 2.3 pixels, respectively. It outperforms **OBJECT** by macro-averages of 4.7%, 3.2%, and 4.2, respectively. In addition, we ran our **Joint** model on a single random fold of the Weizmann horses dataset [6] and achieved 95% pixel accuracy (compared to 89% for **Independent**). This is consistent with the performance of Levin and Weiss [8] and likely near the limit of what methods of this type can achieve.

The results of the **Coordinate** approach described in Section 6 isolate the contribution of the joint inference method that we introduced. This approach was worse than **Joint** by macro-averages of 1%, 1%, and 0.2, demonstrating that the inference routine does in fact contribute to the performance.

## 8 Discussion

This paper presented a new model that fuses methods for object localization and segmentation into a coherent energy model in order to produce more accurate foreground segmentations. The utility of the combined model lies in the use of its outline model in learning the image-specific appearance for the segmentation model, and the terms that encourage agreement between the two while still allowing each the flexibility to reoptimize its own variables. We demonstrated that this model is able to achieve both outlines and segmentations that are superior to several state-of-the-art methods. One promising direction for future work is integration with more sophisticated segmentation algorithms. For example, the use of a robust multi-class segmentation method would allow for class-aware landmark-segment masks that could capture that the giraffe head is often surrounded by sky or trees, while the legs are often found in the grass. Our modular energy function and novel optimization procedure would facilitate such an extension while keeping inference tractable.

**Acknowledgements.** This material is based upon work supported by the National Science Foundation under Grant No. RI-0917151.

## References

1. Heitz, G., Elidan, G., Packer, B., Koller, D.: Shape-based object localization for descriptive classification. In: NIPS (2008)
2. Cootes, T., Edwards, G., Taylor, C.: Active appearance models. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, p. 484. Springer, Heidelberg (1998)
3. Thayananthan, A., Stenger, B., Torr, P., Cipolla, R.: Shape context and chamfer matching in cluttered scenes. In: CVPR (2003)
4. Shotton, J., Blake, A., Cipolla, R.: Contour-based learning for object detection. In: ICCV (2005)
5. Ferrari, V., Jurie, F., Schmid, C.: Accurate object detection with deformable shape models learnt from images. In: CVPR (2007)
6. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: ECCV Workshop on Statistical Learning in Computer Vision (2004)
7. Kumar, M.P., Torr, P., Zisserman, A.: OBJ CUT. In: CVPR (2005)
8. Levin, A., Weiss, Y.: Learning to combine bottom-up and top-down segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 581–594. Springer, Heidelberg (2006)
9. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV (2009)
10. Winn, J., Jojic, N.: Locus: Learning object classes with unsupervised segmentation. In: ICCV (2005)
11. Ramanan, D.: Learning to parse images of articulated objects. In: NIPS (2006)
12. Bray, M., Kohli, P., Torr, P.H.S.: Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 642–655. Springer, Heidelberg (2006)
13. Chen, Y., Zhu, L., Yuille, A.L., Zhang, H.: Unsupervised learning of probabilistic object models (poms) for object classification, segmentation and recognition using knowledge propagation. PAMI (2009)
14. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: CVPR (2004)
15. Fink, M., Ullman, S.: From aardvark to zorro: A benchmark of mammal images. In: IJCV (2008)
16. Torralba, A., Murphy, K., Freeman, W.: Contextual models for object detection using boosted random fields. In: NIPS (2005)
17. 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)
18. Greig, D.M., Porteous, B.T., Seheult, A.H.: Exact maximum a posteriori estimation for binary images. In: Royal Stats. Society (1989)
19. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. In: Royal Stats. Society (1977)
20. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. PAMI (2002)
21. Elidan, G., McGraw, I., Koller, D.: Residual belief propagation: Informed scheduling for async. message passing. In: UAI (2006)
22. Komodakis, N., Paragios, N., Tziritas, G.: Mrf optimization via dual decomposition: Message-passing revisited. In: ICCV (2007)

# SuperParsing: Scalable Nonparametric Image Parsing with Superpixels

Joseph Tighe and Svetlana Lazebnik

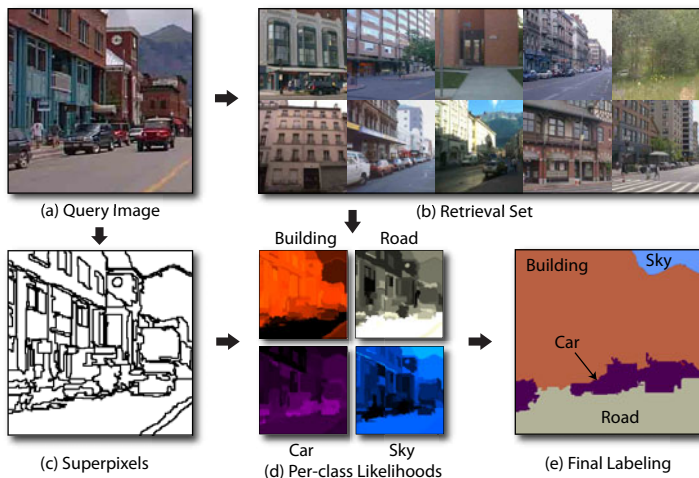
Dept. of Computer Science, University of North Carolina at Chapel Hill,  
Chapel Hill, NC 27599-3175  
{jtighe,lazebnik}@cs.unc.edu

**Abstract.** This paper presents a simple and effective nonparametric approach to the problem of image parsing, or labeling image regions (in our case, superpixels produced by bottom-up segmentation) with their categories. This approach requires no training, and it can easily scale to datasets with tens of thousands of images and hundreds of labels. It works by scene-level matching with global image descriptors, followed by superpixel-level matching with local features and efficient Markov random field (MRF) optimization for incorporating neighborhood context. Our MRF setup can also compute a simultaneous labeling of image regions into semantic classes (e.g., tree, building, car) and geometric classes (sky, vertical, ground). Our system outperforms the state-of-the-art nonparametric method based on SIFT Flow on a dataset of 2,688 images and 33 labels. In addition, we report per-pixel rates on a larger dataset of 15,150 images and 170 labels. To our knowledge, this is the first complete evaluation of image parsing on a dataset of this size, and it establishes a new benchmark for the problem.

**Keywords:** scene understanding, image parsing, image segmentation.

## 1 Introduction

This paper addresses the problem of image parsing, or segmenting all the objects in an image and identifying their categories. The literature contains diverse proposed image parsing methods, including ones that estimate labels pixel by pixel [1,2], ones that aggregate features over segmentation regions [3,4,5,6], and ones that predict object bounding boxes [7,8,9,10]. Most of these methods operate with a few pre-defined classes and require a generative or discriminative model to be trained in advance for each class (and sometimes even for each training exemplar [5]). Training can take days and must be repeated from scratch if new training examples or new classes are added to the dataset. In most cases (with the notable exception of [2]), processing a test image is also quite slow, as it involves operations like running multiple object detectors over the image, performing graphical model inference, or searching over multiple segmentations.



**Fig. 1.** System overview. Given a query image (a) we retrieve similar images from our dataset (b) using several global features. Next, we divide the query into superpixels (c) and compute a per-superpixel likelihood ratio score for each class (d) based on nearest-neighbor superpixel matches from the retrieval set. These scores, in combination with a contextual MRF model, give a dense labeling of the query image (e).

While most existing methods thus remain trapped in a “closed universe” recognition paradigm, a much more exciting paradigm of “open universe” datasets is promising to become dominant in the very near future. For example, the LabelMe dataset [11] is composed of complex, real-world scene images that have been segmented and labeled (sometimes incompletely or noisily) by multiple users. There is no pre-defined set of class labels; the dataset is constantly expanding as people upload new photos or add annotations to current ones. In order to cope with such datasets, vision algorithms must have much faster training and testing times, and they must make it easy to continuously update the visual models with new classes or new images.

Recently, several researchers have begun advocating nonparametric, data-driven approaches to breaking out of the “closed universe” [12,13,14,15]. Such approaches do not do any training at all. Instead, for each new test image, they try to retrieve the most similar training images and transfer the desired information from the training images to the query. Liu et al. [15] have proposed a nonparametric image parsing method based on estimating “SIFT Flow,” or a dense deformation field between images. This method requires no learning and in principle, it can work with an arbitrary set of labels. However, inference via SIFT Flow is currently very complex and computationally expensive. While we agree with [15] that the nonparametric philosophy currently holds the most promise for image parsing in large-scale, dynamic datasets, there is a lot of room for improvement over their method in terms of efficiency.

We set out to implement a nonparametric solution to image parsing that is as straightforward and efficient as possible, and that relies only on operations that can easily scale to ever larger image collections and sets of labels (see Figure 1 for a system overview). Similarly to [15], our proposed method requires no training (just some basic computation of dataset statistics), and makes use of a *retrieval set* of scenes whose content is used to interpret the test image. However, unlike the approach of [15], which works best if the retrieval set images are very similar to the test image in terms of spatial layout of the classes, we transfer labels at the level of *superpixels*, or coherent image regions produced by a bottom-up segmentation method. The label transfer is accomplished with a fast and simple nearest-neighbor search algorithm, and it allows for more variation between the layout of the test image and the images in the retrieval set. Moreover, using segmentation regions as a unit of label transfer gives better spatial support for aggregating features that could belong to the same object [16].

The current consensus among recognition researchers is that image parsing requires *context* (see, e.g., [3,4,9,10]). However, learning and inference with most existing contextual models is slow and non-exact. Therefore, due to our goal of developing a scalable system, we restrict ourselves to efficient forms of context that do not need training and that can be cast in an MRF framework amenable to optimization by fast graph cut algorithms [17,18]. We show that our system equipped with this form of context can achieve results comparable to state-of-the-art systems based on more complex contextual models [3,6]. We also investigate geometric/semantic context in the manner of Gould et al. [6]. Namely, for each superpixel in the image, we simultaneously estimate a semantic label (e.g., building, car, person, etc.) and a geometric label (sky, ground, or vertical surface) while enforcing coherence between the two class types.

Our system exceeds the results reported in [15] on a dataset of 2,688 images and 33 labels. Moreover, to demonstrate the scalability of our method, we present per-pixel and per-class rates on a subset of LabelMe with 15,150 images and 170 labels. To our knowledge, we are the first to report complete recognition results on a dataset of this size. Thus, one of the contributions of this work is to establish a new benchmark for large-scale image parsing. Our code, data, and output can be found at <http://www.cs.unc.edu/SuperParsing>.

## 2 System Description

### 2.1 Retrieval Set

Similarly to several other data-driven methods [7,12,14,15], our first step in parsing a query test image is to find a relatively small *retrieval set* of training images that will serve as the source of candidate superpixel-level annotations. This is done not only for computational efficiency, but also to provide scene-level context for the subsequent superpixel matching step. A good retrieval set should contain images of a similar scene type to that of the test image, along with similar objects and spatial layouts. To attempt to indirectly capture this kind of similarity, we use four types of global image features (Table 1(a)): spatial

**Table 1.** A complete list of features used in our system

(a) Global features for retrieval set computation (Section 2.1)		
Type	Name	Dimension
Global	Spatial pyramid (3 levels, SIFT dictionary of size 200)	4200
	Gist (3-channel RGB, 3 scales with 8, 8, & 4 orientations)	960
	Tiny image (3-channel RGB, $16 \times 16$ pixels)	768
	Color histogram (3-channel RGB, 8 bins per channel)	24
(b) Superpixel features (Section 2.2)		
Shape	Mask of superpixel shape over its bounding box ( $8 \times 8$ )	64
	Bounding box width/height relative to image width/height	2
	Superpixel area relative to the area of the image	1
Location	Mask of superpixel shape over the image	64
	Top height of bounding box relative to image height	1
Texture/SIFT	Texton histogram, dilated texton histogram	$100 \times 2$
	SIFT histogram, dilated SIFT histogram	$100 \times 2$
	Left/right/top/bottom boundary SIFT histogram	$100 \times 4$
Color	RGB color mean and std. dev.	$3 \times 2$
	Color histogram (RGB, 11 bins per channel), dilated hist.	$33 \times 2$
Appearance	Color thumbnail ( $8 \times 8$ )	192
	Masked color thumbnail	192
	Grayscale gist over superpixel bounding box	320

pyramid [19], gist [20], tiny image [13], and color histogram. For each feature type, we rank all training images in increasing order of Euclidean distance from the query. Then we take the minimum of the per-feature ranks to get a single ranking for each training image, and take the top 200 images as the retrieval set. Taking the minimum of per-feature ranks amounts to taking the top fifty matches according to each global image descriptor, and it gives us better results than, say, averaging the ranks. Intuitively, taking the best scene matches from each of the global descriptors leads to better superpixel-based matches for region-based features that capture similar types of cues as the global features.

## 2.2 Superpixel Features

We wish to label the query image based on the content of the retrieval set, but assigning labels on a per-pixel basis as in [11,14,15] would be too inefficient. Instead, like [3,4,5], we choose to assign labels to superpixels, or regions produced by bottom-up segmentation. This not only reduces the complexity of the problem, but also gives better spatial support for aggregating features that could belong to a single object than, say, fixed-size square patches centered on every pixel in the image. We obtain superpixels using the fast graph-based segmentation algorithm of [21] and describe their appearance using 20 different features similar to those of [5], with some modifications and additions. A complete list of the features is given in Table 1(b). In particular, we compute histograms of



textons and dense SIFT descriptors over the superpixel region, as well as that region dilated by 10 pixels. For SIFT features, which are more powerful than textons, we have also found it useful to compute left, right, top, and bottom boundary histograms. To do this, we find the boundary region as the difference between the superpixel dilated and eroded by 5 pixels, and then obtain the left/right/top/bottom parts of the boundary by cutting it with an “X” drawn over the superpixel bounding box. All of the features are computed for each superpixel in the training set and stored together with their class labels. We associate a class label with a training superpixel if 50% or more of the superpixel overlaps with the segment mask for that label.

### 2.3 Local Superpixel Labeling

Having segmented the test image and extracted all its features, we next obtain a likelihood ratio score for each test superpixel and each class that is present in the retrieval set. Making the Naive Bayes assumption that features are independent of each other given the class, the likelihood ratio for class  $c$  and superpixel  $s_i$  is

$$L(s_i, c) = \frac{P(s_i|c)}{P(s_i|\bar{c})} = \prod_k \frac{P(f_i^k|c)}{P(f_i^k|\bar{c})}, \quad (1)$$

where  $\bar{c}$  is the set of all classes excluding  $c$ , and  $f_i^k$  is the feature vector of the  $k$ th type for  $s_i$ . Each likelihood ratio  $P(f_i^k|c)/P(f_i^k|\bar{c})$  is computed with the help of nonparametric density estimates of features from the required class(es) in the neighborhood of  $f_i^k$ . Specifically, let  $\mathcal{D}$  denote the set of all superpixels in the training set, and  $\mathcal{N}_i^k$  denote the set of all superpixels in the retrieval set whose  $k$ th feature distance from  $f_i^k$  is below a fixed threshold  $t_k$ . Then we have

$$\frac{P(f_i^k|c)}{P(f_i^k|\bar{c})} = \frac{n(c, \mathcal{N}_i^k)/n(c, \mathcal{D})}{n(\bar{c}, \mathcal{N}_i^k)/n(\bar{c}, \mathcal{D})} = \frac{n(c, \mathcal{N}_i^k)}{n(\bar{c}, \mathcal{N}_i^k)} \times \frac{n(\bar{c}, \mathcal{D})}{n(c, \mathcal{D})}, \quad (2)$$

where  $n(c, \mathcal{S})$  (resp.  $n(\bar{c}, \mathcal{S})$ ) is the number of superpixels in set  $\mathcal{S}$  with class label  $c$  (resp. not  $c$ ). To prevent zero likelihoods and smooth the counts, we add one to  $n(c, \mathcal{N}_i^k)$  and  $n(\bar{c}, \mathcal{N}_i^k)$ . In our implementation, we use the  $\ell_2$  distance for all features, and set each threshold  $t_k$  to the median distance to the 20th nearest neighbor for the  $k$ th feature type over the dataset. The superpixel neighbors  $\mathcal{N}_i^k$  are currently found by linear search through the retrieval set.

At this point, we can obtain a labeling of the image by simply assigning to each superpixel the class that maximizes eq. (1). As shown in Table 2, the resulting classification rates already come within 1.5% of those of [15]. We are not aware of any comparably simple scoring scheme reporting such encouraging results for image parsing problems with many unequally distributed labels.

### 2.4 Contextual Inference

Next, we would like to enforce contextual constraints on the image labeling – for example, a labeling that assigns “water” to a superpixel completely surrounded by “sky” is not very plausible. Many state-of-the-art approaches encode

such constraints with the help of conditional random field (CRF) models [16,4]. However, CRFs tend to be very costly both in terms of learning and inference. In keeping with our nonparametric philosophy and emphasis on scalability, we restrict ourselves to contextual models that require minimal training and that can be solved efficiently. Therefore, we formulate the global image labeling problem as minimization of a standard MRF energy function defined over the field of superpixel labels  $\mathbf{c} = \{c_i\}$ :

$$J(\mathbf{c}) = \sum_{s_i \in SP} E_{\text{data}}(s_i, c_i) + \lambda \sum_{(s_i, s_j) \in A} E_{\text{smooth}}(c_i, c_j), \quad (3)$$

where  $SP$  is the set of superpixels,  $A$  is the set of pairs of adjacent superpixels and  $\lambda$  is the smoothing constant. We define the data term as  $E_{\text{data}} = -w_i \log L(s_i, c_i)$ , where  $L(s_i, c_i)$  is the likelihood ratio score from eq. (1) and  $w_i$  is the superpixel weight (the size of  $s_i$  in pixels divided by the mean superpixel size). The smoothing term  $E_{\text{smooth}}$  is defined based on probabilities of label co-occurrence:

$$E_{\text{smooth}}(c_i, c_j) = -\log[(P(c_i|c_j) + P(c_j|c_i))/2] \times \delta[c_i \neq c_j], \quad (4)$$

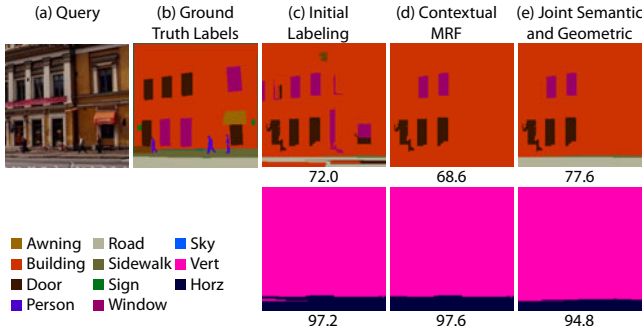
where  $P(c|c')$  is the conditional probability of one superpixel having label  $c$  given that its neighbor has label  $c'$ , estimated by counts from the training set. We use the two conditionals  $P(c|c')$  and  $P(c'|c)$  instead of the joint  $P(c, c')$  because they have better numerical scaling, and average them to obtain a symmetric quantity. Qualitatively, we have found eq. (4) to produce very reasonable edge penalties. As can be seen from the examples in Figure 4 (d) and (f), it successfully flags improbable boundaries between “sea” and “sun,” and “mountain” and “building.” Quantitatively, results with eq. (4) tend to be about 1% more accurate than with the constant Potts penalty  $\delta[c_i \neq c_j]$ . We perform MRF inference using the efficient graph cut optimization code of [17,18,22]. On our large datasets, the resulting global labelings improve the accuracy by 3-5% (Table 2).

## 2.5 Simultaneous Classification of Semantic and Geometric Classes

Following Gould et al. [6], we consider the task of simultaneously labeling regions into two types of classes: semantic and geometric. Like [6], we use three geometric labels – sky, ground, and vertical – although the sets of semantic labels in our datasets are much larger. In this paper, we make the reasonable assumption that each semantic class is associated with a unique geometric class (e.g., “building” is “vertical,” “river” is “horizontal,” and so on) and specify this mapping by hand. We jointly solve for the fields of semantic labels ( $\mathbf{c}$ ) and geometric labels ( $\mathbf{g}$ ) by minimizing a cost function that is a simple extension of eq. (4):

$$H(\mathbf{c}, \mathbf{g}) = J(\mathbf{c}) + J(\mathbf{g}) + \mu \sum_{s_i \in SP} \varphi(c_i, g_i), \quad (5)$$

where  $\varphi$  is the term that enforces coherence between the geometric and semantic labels. It is 0 when the semantic class  $c_i$  is of the geometric class type  $g_i$  and



**Fig. 2.** In the contextual MRF classification, the road gets replaced by “building,” while “horizontal” is correctly classified. By jointly solving for the two kinds of labels, we manage to recover some of the “road” and “sidewalk” in the semantic labeling. Note also that in this example, our method correctly classifies some of the windows that are mislabeled as doors in the ground truth, and incorrectly but plausibly classifies the windows on the lower level as doors.

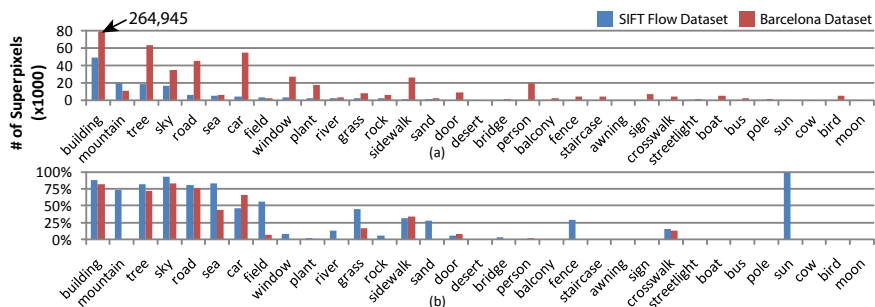
1 otherwise. The constant  $\mu$  controls how strictly the coherence is enforced (we use  $\mu = 8$  in all experiments). Note that we can enforce the semantic/geometric consistency in a hard manner by effectively setting  $\mu = \infty$ , but we have found that allowing some tradeoff produces better results. Eq. (5) is in a form that can be optimized by the  $\alpha/\beta$ -swap algorithm [17,18,22]. The inference takes almost the same amount of time as for the MRF setup of the previous section. Figure 2 shows an example where joint inference over semantic and geometric labels improves the accuracy of the semantic labeling. In many other cases, joint inference improves both labelings.

## 3 Results

### 3.1 Large Datasets

The first large-scale dataset in our experiments (“SIFT Flow dataset” in the following) is composed of the 2,688 images that have been thoroughly labeled by LabelMe users. Liu *et al.* [15] have split this dataset into 2,488 training images and 200 test images and used synonym correction to obtain 33 semantic labels. In our experiments, we use the same training/test split as [15]. Our second dataset (“Barcelona” in the following) is derived from the LabelMe subset used in [7]. It has 14,871 training and 279 test images.<sup>1</sup> The test set consists of street scenes from Barcelona, while the training set ranges in scene type but has no street scenes from Barcelona. We manually consolidated the synonyms in the

<sup>1</sup> Russell *et al.* [7] use a test set of 560 images, 281 of which are office scenes. However, the entire training set of 14,871 images only contains about 218 office scenes, so we have excluded the office images from the test set.



**Fig. 3.** (a) Label frequencies for the superpixels in the training set. The Barcelona dataset has 170 labels, but we only show the ones that are in common with the SIFT Flow dataset. (b) Per-class classification rates of our system.

label set to 170 unique labels. Note that [7] only gives detection curves for 12 categories on this dataset, so there are no previous baseline results for per-pixel performance. Both datasets have very nonuniform label distributions, as shown in Figure 3(a). Because of this, we report not only the per-pixel classification rate, which mainly reflects performance on the few largest classes, but also the average of per-pixel rates of all the classes.

Our system labels each superpixel of each test image by a *semantic class* (the original 33 and 170 labels, respectively) and a *geometric class* of sky, ground, or vertical (same as [6]). Because the number of geometric classes is small and fixed for all datasets, we have trained a discriminative model for them using a boosted decision tree classifier as in [3]. This classifier outputs a likelihood ratio score that we can directly plug into our MRF framework, and it gives us an improvement of about 1% in the accuracy for geometric classes over the nearest-neighbor scheme of Section 2.3. Apart from this, local and contextual MRF classification for geometric classes proceeds as described in Sections 2.3 and 2.4, and we also put the geometric and semantic likelihood ratios into a joint contextual classification framework as described in Section 2.5.

Table 2 reports per-pixel and average per-class rates for semantic classification of all three setups (local superpixel labeling, contextual MRF, joint MRF). As compared to the local baseline, the contextual MRF improves overall per-pixel rates on the SIFT Flow dataset by about 3% and on the Barcelona dataset by about 4%. Average per-class rates drop slightly due to the MRF “smoothing away” some of the smaller classes. Simultaneous geometric/semantic MRF improves the results for both types of classes on the SIFT Flow dataset, but makes little difference on the Barcelona dataset. Figure 3(b) shows that our per-class rates on both datasets are comparable, with large changes due primarily to differences in label frequency (e.g., there are no mountains in the Barcelona test set). It also shows that, similarly to most other image labeling approaches that do not train object detectors, we get much weaker performance on “things” (people, cars, signs) than on “stuff” (sky, road, trees).

**Table 2.** Performance of our system on the two large datasets. For semantic classes, we show the per-pixel rate followed by the average per-class rate in parentheses. Because there are only three geometric classes, we report only the per-pixel rate for them.

	SIFT Flow dataset [15]		Barcelona dataset [7]	
	Semantic	Geometric	Semantic	Geometric
Baseline	74.75 [15]	N/A	N/A	N/A
Local labeling (Sec. 2.3)	73.2 (29.1)	89.8	62.5 (8.0)	89.9
Superpixel MRF (Sec. 2.4)	76.3 (28.8)	89.9	66.6 (7.6)	90.2
Simultaneous MRF (Sec. 2.5)	76.9 (29.4)	90.8	66.9 (7.6)	90.7

Our final system on the SIFT Flow dataset achieves a classification rate of 76.9%. Thus, we outperform Liu *et al.* [15], who report a rate of 74.75% on the same test set with a more complex pixel-wise MRF (without the pixel-wise MRF, their rate is 66.24%). Liu *et al.* [15] also cite a rate of 82.72% for the top seven object categories; our corresponding rate is 84.5%. Sample output of our system on several SIFT Flow test images can be seen in Figure 4.

Next, we examine the effects of various components of our system. For each of these tests, we only show the local labeling rates on the SIFT Flow dataset. Table 3(a) shows the effect of different combinations of global features for computing the retrieval set (Section 2.1). Similarly to [12], we find that combining global features of unequal descriptive power gives better scene matches. Table 3(b) shows classification rates of the system with ten superpixel features added consecutively in decreasing order of their contribution to performance. Notice that SIFT histograms constitute four of the top ten features selected. The dilated SIFT histogram, which already incorporates some context from the superpixel neighborhood, is our single strongest feature, and it effectively makes the non-dilated SIFT histogram redundant. Also notice that SIFT and texton histograms are complementary (despite SIFT being stronger), and that all six feature categories from Table 1(b) are represented in the top ten.

Table 4(a) examines the effect of retrieval set size on classification rate. Interestingly, matching test superpixels against the entire dataset (last row of the table) drastically reduces performance. Thus, we quantitatively confirm the intuition that the retrieval set is not just a way to limit the computational complexity of sub-image matching; it acts as a global image-level context by restricting the superpixel matches to come from a small subset of related scenes. Table 4(b) shows the effect of restricting the list of possible labels in a test image to different “shortlists.” Effectively, the shortlist used by our system for each test image is composed of all the classes present in the retrieval set (first row). To demonstrate the effect of long-tail class frequencies, the second row of the table shows the performance we get by classifying every superpixel in every test image to the ten most common classes in dataset. This does not change the overall per-pixel rate, but lowers the average per-class rate dramatically, thus underscoring the importance of looking at both numbers. The third row of Table 4(b) shows the results produced by restricting our shortlist to the ground truth labels in the

**Table 3.** Feature evaluation on the SIFT Flow dataset. (a) Results for local superpixel labeling with different retrieval set feature combinations. (b) Performance with the best retrieval set from (a) and top ten superpixel features added in succession.

Global Descriptor	Rate
Gist (G)	70.8 (28.7)
Spatial Pyramid (SP)	70.0 (22.4)
Color Hist. (CH)	65.9 (22.1)
Tiny Image (TI)	65.4 (25.5)
G + SP	72.4 (27.6)
G + SP + CH	73.3 (28.8)
G + SP + CH + TI	73.3 (29.1)

Superpixel Feature	Rate
Dilated SIFT hist.	44.8 (20.8)
+ Texton hist.	54.3 (21.1)
+ Top height	60.2 (23.2)
+ Color thumbnail	63.6 (25.0)
+ Dilated color hist.	66.4 (26.1)
+ Left boundary SIFT hist.	68.1 (26.8)
+ Right boundary SIFT hist.	69.4 (26.3)
+ SP mask over bounding box	69.8 (27.3)
+ Top boundary SIFT hist.	70.5 (27.9)
+ Color hist.	71.0 (27.9)
+ All remaining features	73.3 (29.1)

query image, giving us an upper bound for the performance of superpixel matching. We can see that a perfect shortlist “oracle” would give us a boost of almost 8%. This suggests that to further improve system performance, we may get a bigger payoff from more accurate scene-level label prediction, rather than from more sophisticated edge potentials in the MRF. In fact, we have observed that in many of our unsuccessfully labeled images, incompatible scene classes with strong local support over large regions vie for the interpretation of the image, and neighborhood context, though it may detect the conflict, has no plausible path towards resolving it (Figure 4(f) is one example of this).

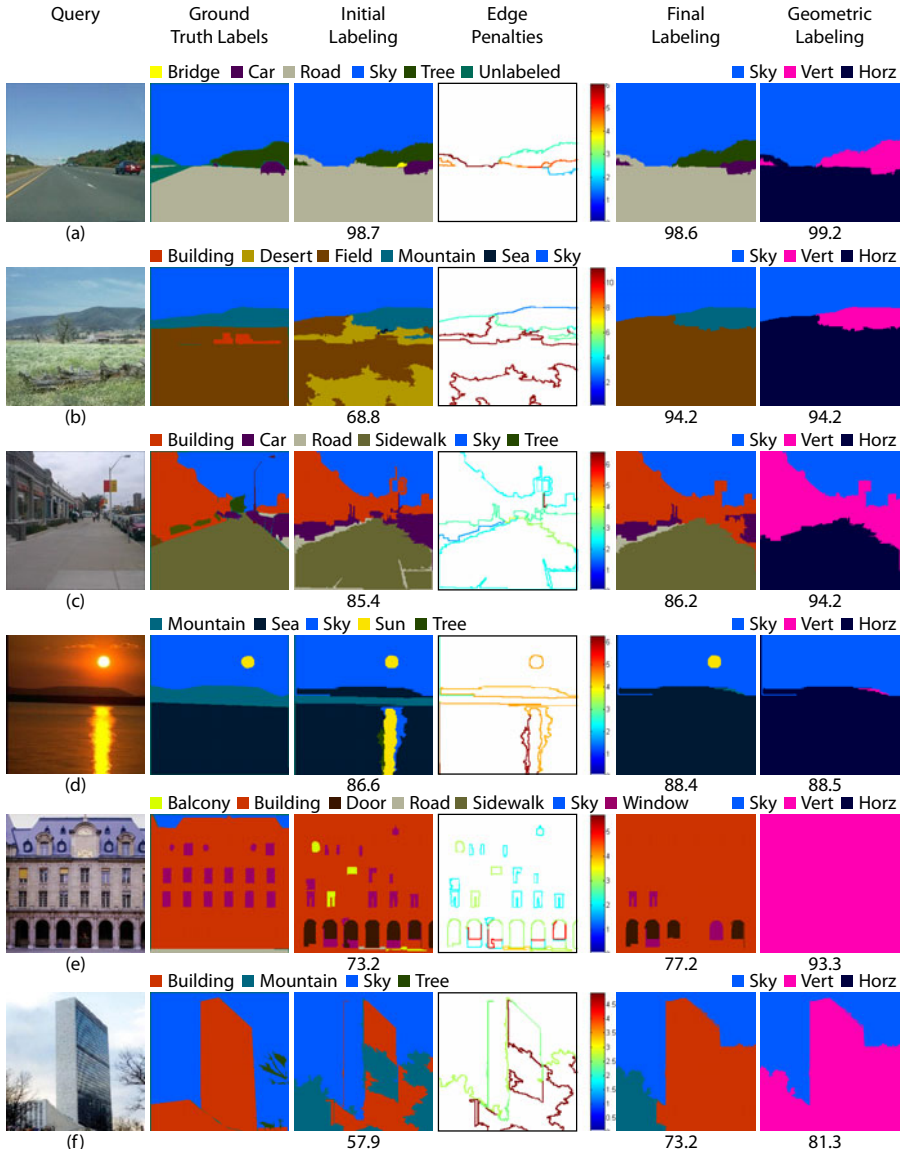
Finally, we analyze the computational requirements of our system. Our current implementation is mostly in unoptimized and un-parallelized MATLAB (with some outside C code for feature extraction and MRF optimization), and all our tests are run on a single PC with dual Xeon 2.33 GHz quad core processors and 24 GB RAM. Table 5 shows a breakdown of the main stages of the computation. On the SIFT Flow dataset, we are able to extract features and label images in less than 10 seconds. In comparison, as reported in [15], to classify a single query image, the SIFT Flow system required 50 alignment operations that took 30 seconds each, or 25 minutes total without parallelization.

**Table 4.** (a) Effect of retrieval set size on performance for the SIFT Flow dataset. (b) Effect of restricting the set of possible classes in the test image to different “shortlists.”

Retrieval Set Size	Rate
50	71.1 (30.1)
100	72.4 (29.7)
200	73.3 (29.1)
400	72.1 (27.2)
2,488	68.6 (19.1)

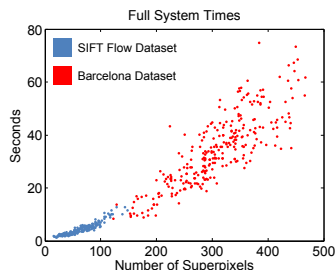
Shortlist	Rate
Classes in retrieval set	73.3 (29.1)
10 most common classes	73.2 (20.4)
Perfect shortlist	81.0 (34.0)



**Fig. 4.** Example results from the SIFT Flow test set (best viewed in color). In (c), sidewalk is successfully recovered. In (d), the co-occurrence MRF and joint geometric/semantic classification remove the spurious classification of the sun’s reflection in the water as “sun.” In (e), we find some windows (some of which are smoothed away by the MRF) and plausibly classify the arches at the bottom of the building as doors. In (f), parts of the building and the bare tree get initially classified as “mountain,” and while the co-occurrence MRF does not like the boundaries between “building” and “mountain,” it is not completely successful in eliminating the errors. For complete results, see <http://www.cs.unc.edu/SuperParsing>.

**Table 5.** Left: The average timing in seconds of the different stages in our system (excluding file I/O). While the runtime is significantly longer for the Barcelona dataset, this is primarily due to the change in image size and not the number of images. Right: query time vs. number of superpixels in the query image.

	SIFT Flow	Barcelona
Training set size	2,488	14,871
Image size	256 × 256	640 × 480
Ave. # superpixels	63.9	307.9
Feature extraction	~ 4 sec	~ 5 min
Retrieval set search	0.04 ± 0.0	0.21 ± 0.0
Superpixel search	4.4 ± 2.3	34.2 ± 13.4
MRF solver	0.005 ± 0.003	0.03 ± 0.02
Total (excluding features)	4.4 ± 2.3	34.4 ± 13.4



At present, our running time is actually dominated by our (wildly inefficient) feature extraction code that can be easily sped up by an order of magnitude. Our algorithm complexity is approximately quadratic in the average number of superpixels per image in the dataset due to the need to exhaustively match every test superpixel to every retrieval set superpixel. On the other hand, this time is independent of the overall number of training images. Moreover, as our dataset gets larger, we expect that target retrieval set size will stay the same or decrease, as the top scene matches will become closer to the test image. For larger datasets, the main bottleneck of our system will not be superpixel search, but retrieval set search and file I/O for loading retrieval set superpixel descriptors from disk. However, we expect to be able to overcome these challenges with appropriate hardware, parallelization, and/or data structures for fast search.

### 3.2 Small Datasets

To further validate our superpixel-based feature representation, we tested it on two small datasets: that of Gould et al. [6], which has 715 images with eight semantic and three geometric classes, and the geometric context dataset of Hoiem et al. [3], which has 300 images and seven surface layout classes (sky, ground, and five vertical sub-classes). For the latter, we treat these seven classes as the “semantic” classes, and the three geometric classes correspond to the main classes of [3]. Because nearest neighbor search requires a large set of training images to perform well, and because the competing approaches use heavily trained discriminative models, we train boosted decision tree classifiers similar to those of [3] on all the semantic and geometric classes. To obtain initial labelings of test images in these datasets, we do not use a retrieval set, but apply the boosted tree classifier for each class to each superpixel and use its likelihood ratio score in the same way as eq. (1). Table 6 shows the resulting performance, which is comparable to the results of [3,6]. Moreover, our system is much simpler than the competing approaches. Unlike [6], we do not need to learn classifiers over pairs of geometric and semantic classes or optimize image regions in a complex CRF



**Table 6.** A comparison of our system to [6,3] using five-fold cross-validation and the same evaluation protocols as [6,3]

	Gould <i>et al.</i> dataset [6]		Geometric Context dataset [3]	
	Semantic	Geometric	Sub-classes	Main classes
Gould <i>et al.</i> [6]	76.4	91.0	N/A	86.9
Hoiem <i>et al.</i> [3]	N/A	N/A	61.5	88.1
Local labeling	76.9	90.5	57.6	87.8
Superspixel MRF	77.5	90.6	61.0	88.2
Simultaneous	77.5	90.6	61.0	88.1

framework. Unlike [3], we do not need to search over multiple segmentations with two tiers of features and a discriminative model of region homogeneity. In fact, when restricted to just a single superspixel segmentation, Hoiem *et al.* [3] report a sub-class rate of 53.5%, which we beat by 7.5% on the same superspixels.

## 4 Discussion

This paper has presented a superspixel-based approach to image parsing that can take advantage of datasets consisting of tens of thousands of images annotated with hundreds of labels. Our system does not need training, except for basic computation of dataset statistics such as label co-occurrence probabilities, and it relies on just a few constants that are kept fixed for all datasets. Our experimental evaluation carefully justifies every implementation choice. Despite its simplicity, our system outperforms state-of-the-art methods such as SIFT Flow [15]. Like [15], our method is nonparametric and makes use of a retrieval set of similar scenes, but unlike [15], it does not rely on an intricate optical flow-like scene alignment model. Our underlying feature representation, based on multiple appearance descriptors computed over segmentation regions, is similar to that of [3,5]. However, unlike [3], we do not search over multiple segmentations, and unlike [5], we successfully combine features without learning class-specific or exemplar-specific distance functions. That one can achieve good performance without these costly steps is very encouraging for the prospect of successfully scaling up image parsing algorithms.

There still remain areas to improve our system further. Because our representation makes it easy to “plug in” new features, any advances in feature extraction are likely to give gains in performance. Also, while we have achieved promising results with one bottom-up segmentation algorithm [21], it remains important to examine the effect of segmentation quality on image parsing and to address the problem of finding the right spatial support for objects.

## Acknowledgments

This research was supported in part by NSF CAREER award IIS-0845629, Microsoft Research Faculty Fellowship, and Xerox.

## References

1. He, X., Zemel, R., Carreira-Perpinan, M.: Multiscale CRFs for image labeling. In: CVPR (2004)
2. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: CVPR (2008)
3. Hoiem, D., Efros, A., Hebert, M.: Recovering surface layout from an image. *IJCV* 75, 151–172 (2007)
4. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in context. In: ICCV, Rio de Janeiro (2007)
5. Malisiewicz, T., Efros, A.A.: Recognition by association via learning per-exemplar distances. In: CVPR (2008)
6. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV (2009)
7. Russell, B.C., Torralba, A., Liu, C., Fergus, R., Freeman, W.T.: Object recognition by scene alignment. In: NIPS (2007)
8. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multi-scale, deformable part model. In: CVPR (2008)
9. Heitz, G., Koller, D.: Learning spatial context: Using stuff to find things. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 30–43. Springer, Heidelberg (2008)
10. Divvala, S.K., Hoiem, D., Hays, J.H., Efros, A.A., Hebert, M.: An empirical study of context in object detection. In: CVPR (2009)
11. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: LabelMe: a database and web-based tool for image annotation. *IJCV* 77, 157–173 (2008)
12. Hays, J., Efros, A.A.: Im2gps: Estimating geographic information from a single image. In: CVPR (2008)
13. Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: a large dataset for non-parametric object and scene recognition. *PAMI* 30, 1958–1970 (2008)
14. Liu, C., Yuen, J., Torralba, A., Sivic, J., Freeman, W.T.: SIFT flow: dense correspondence across difference scenes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 28–42. Springer, Heidelberg (2008)
15. Liu, C., Yuen, J., Torralba, A.: Nonparametric scene parsing: label transfer via dense scene alignment. In: CVPR (2009)
16. Gu, C., Lim, J.J., Arbelaez, P., Malik, J.: Recognition using regions. In: CVPR (2009)
17. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? *PAMI* 26, 147–159 (2004)
18. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI* 26, 1124–1137 (2004)
19. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2006)
20. Oliva, A., Torralba, A.: Building the gist of a scene: The role of global image features in recognition. *Visual Perception, Progress in Brain Research* 155 (2006)
21. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. *International Journal of Computer Vision* 2 (2004)
22. Bagon, S.: Graph cut matlab wrapper (2006), <http://www.wisdom.weizmann.ac.il/~bagon>

# Segmenting Salient Objects from Images and Videos

Esa Rahtu<sup>1</sup>, Juho Kannala<sup>1</sup>, Mikko Salo<sup>2</sup>, and Janne Heikkilä<sup>1</sup>

<sup>1</sup> Machine Vision Group, University of Oulu, Finland

<sup>2</sup> Department of Mathematics and Statistics, University of Helsinki, Finland

**Abstract.** In this paper we introduce a new salient object segmentation method, which is based on combining a saliency measure with a conditional random field (CRF) model. The proposed saliency measure is formulated using a statistical framework and local feature contrast in illumination, color, and motion information. The resulting saliency map is then used in a CRF model to define an energy minimization based segmentation approach, which aims to recover well-defined salient objects. The method is efficiently implemented by using the integral histogram approach and graph cut solvers. Compared to previous approaches the introduced method is among the few which are applicable to both still images and videos including motion cues. The experiments show that our approach outperforms the current state-of-the-art methods in both qualitative and quantitative terms.

**Keywords:** Saliency measure, background subtraction, segmentation.

## 1 Introduction

Biological vision systems are remarkably effective in finding relevant targets from a scene [1]. Identifying these prominent, or salient, areas in the visual field enables one to allocate the limited perceptual resources in an efficient way. Compared to biological systems, computer vision methods are far behind in the ability of saliency detection. However, reliable saliency detection methods would be useful in many applications like adaptive compression and scaling [2,3], unsupervised image segmentation [4,5], and object recognition [6,7].

Perhaps the most common approach to reduce scene clutter is to detect moving objects against a static background [8,9,10]. These methods have been very successful in many applications, but they have severe limitations in the case of dynamic scenes or moving cameras. These circumstances have been addressed by introducing adaptive background models and methods to eliminate camera movements [11,12], but both of these are difficult problems and technically demanding. Moreover, the methods in this class are applicable only to video sequences, but not to still images where motion cues are not available.

A different approach is provided by supervised object detection techniques, which are aimed at finding particular categories like persons, tables, cars, etc. [13,14,15]. These methods have resulted in high performance, but the limitation



**Fig. 1.** Example result achieved using the proposed approach. From left to right: original image, saliency map, segmentation by thresholding, and segmentation by using the CRF model.

is that the objects of interest must reside in the predefined categories from which the training samples must be available. Furthermore, the training process is rather extensive and the performance is dictated by the training data.

An alternative method is offered by general purpose saliency detectors. These methods are inspired by the ability of human visual system to quickly focus on general salient targets without preceding training. Such techniques are suitable in situations where possible targets and imaging conditions are not known in advance. Perhaps the first biologically plausible saliency detector was presented in [16], where the key idea was based on contrast measurements using difference of Gaussians filtering.

Since [16] several saliency detectors have been introduced. They are similarly focusing on estimating local feature contrast between image regions and their surroundings. Most methods implement this by local filtering or sliding window techniques [18,19,20,21,22]. Other methods apply Fourier transform [23,24], mutual information formulation [25], or band-pass filtering [26].

The main limitation with many general saliency detection methods is their low resolution, e.g.  $64 \times 64$  with the approaches in [23,24] and small fraction of the image dimension with [16,17]. An exception to this is provided by sliding window based methods [20,21,22] and the band-pass filtering approach [26], where the output map has the same resolution as the input image. Another drawback is that only few methods [22,24] are capable of incorporating motion cues in the saliency map. Finally, large computational demands and variable parameters are limiting the usage of several methods [16,18,19,22].

In the previous experiments the sliding window and band-pass filtering approaches have resulted in the best performance [26]. Based on this observation we present a new saliency segmentation method, which is a composition of a sliding window based saliency measure and a conditional random field (CRF) segmentation model. The introduced saliency measure is based on a rigorous statistical formulation enabling feature level information fusion and analysis of the robustness properties.

In contrast to previous methods our approach is directly applicable to both still images and videos including also motion cues in the saliency measure and the

CRF model. The method which is the most similar to our saliency measure is the approach in [21], but it differs in the formulation of saliency measure, information fusion approach, and application of motion cues in estimation. Experiments with the saliency segmentation test framework [26] show considerable improvements in terms of both precision and recall. Current state-of-the-art methods are outperformed slightly even by using a simple thresholding of the saliency map, and with a clear margin when the proposed CRF model is used.

**Contributions.** We present a salient object segmentation method for images and video sequences. The contributions of our paper include:

1. A rigorous statistical formulation of a saliency measure, which is based on local feature contrast, and analysis of its properties under noisy data.
2. Feature level information fusion in the construction of saliency maps and inclusion of motion cues by using optical flow.
3. CRF model for segmenting objects in images and videos based on information in saliency maps.

## 2 Saliency Measure

In this section we describe the proposed saliency measure. The measure is based on applying a sliding window to the image, and on comparing in each window the contrast between the distribution of certain features in an inner window to the distribution in the collar of the window. The basic setup for this saliency measure was introduced in [21], but here we will modify it by taking into account the properties 1 and 2 listed in the contributions above.

### 2.1 Definition of Saliency Measure

Consider an image in  $\mathbb{R}^2$  and a map  $F$  which maps every point  $x$  to a certain feature  $F(x)$  (which could be the intensity, the value in different color channels, or information obtained from motion). The feature space is divided into disjoint bins, with  $Q_{F(x)}$  denoting the bin which contains  $F(x)$ .

We consider a rectangular window  $W$  divided into two disjoint parts, a rectangular inner window  $K$  (the kernel) and the border  $B$  (see Figure 2), and apply the hypothesis that points in  $K$  are salient and points in  $B$  are part of the background. A similar hypothesis has also been used in [21, 22]. Let  $Z$  be a random variable with values in  $W$ , describing the distribution of pixels in  $W$ . Under the stated hypothesis, the saliency measure of a point  $x \in K$  is defined to be the conditional probability

$$S_0(x) = P(Z \in K | F(Z) \in Q_{F(x)}). \quad (1)$$

The saliency measure of  $x$  is always a number between 0 and 1. It follows from the definition that a pixel  $x$  is salient (that is,  $S_0(x)$  is close to 1) if the feature at  $x$  is similar to the features at points of the inner window (and different from points in the border).

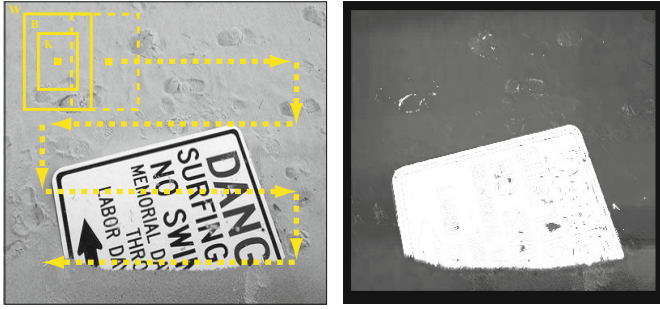


Fig. 2. Illustration of saliency map computation

The computation of  $S_0(x)$  can be achieved through the Bayes formula  $P(A|B) = P(B|A)P(A)/P(B)$ . Using the abbreviations  $H_0$ ,  $H_1$ , and  $F(x)$  for the events  $Z \in K$ ,  $Z \in B$ , and  $F(Z) \in Q_{F(x)}$ , respectively, gives that

$$S_0(x) = \frac{P(F(x)|H_0)P(H_0)}{P(F(x)|H_0)P(H_0) + P(F(x)|H_1)P(H_1)}. \quad (2)$$

The computation of this measure is greatly simplified if we assume that  $Z$  has a probability density function  $p$  which is constant on  $K$  and on  $B$ . In fact, given  $p_0$  with  $0 < p_0 < 1$ , we take  $p(x) = p_0/|K|$  for  $x \in K$  and  $p(x) = (1 - p_0)/|B|$  for  $x \in B$ . With this choice, the conditional probabilities in the last expression for  $S_0(x)$  become normalized histograms. For instance, for the set  $K$  we write

$$h_K(x) = P(F(x)|H_0) = \frac{1}{P(H_0)} \int_{K \cap F^{-1}(Q_{F(x)})} p(w) dw. \quad (3)$$

Since  $p$  is constant on  $K$ , the discretized version of the last quantity is obtained by just counting the number of points  $z$  in  $K$  for which  $F(z)$  is in  $Q_{F(x)}$ , and by dividing by the number of points in  $K$ . Defining similarly  $h_B(x) = P(F(x)|H_1)$ , the saliency measure may be written as

$$S_0(x) = \frac{h_K(x)p_0}{h_K(x)p_0 + h_B(x)(1 - p_0)}. \quad (4)$$

Clearly  $S_0(x)$  is always a number between 0 and 1.

## 2.2 Regularized Saliency Measure

Note that a small change in the function  $F$  may change the bin of  $F(x)$ , possibly resulting in a large change in the value of  $h_K(x)$ . Therefore the measure  $S_0(x)$  is not stable with respect to noise. To increase robustness we introduce a regularized saliency measure. For computational purposes it is most convenient to regularize the normalized histograms directly.

Assume that the bins in feature space are indexed by integers  $j$ , and let  $j(x)$  be such that  $F(x)$  lies in the bin  $Q_{j(x)}$ . Let also  $h_A(j) = h_A(x)$  for  $j = j(x)$ . If  $\alpha > 0$  let  $g_\alpha(x) = c_\alpha e^{-\frac{x^2}{2\alpha}}$  be the Gaussian function with variance  $\alpha$ , normalized so that  $\sum_{j=-\infty}^{\infty} g_\alpha(j) = 1$ . Define the regularized histogram  $h_{K,\alpha}(j) = \sum_{k=-\infty}^{\infty} g_\alpha(j - k)h_K(k)$ . With a similar definition for  $h_{B,\alpha}$ , the regularized saliency measure is defined by

$$S_\alpha(x) = \frac{h_{K,\alpha}(j(x))p_0}{h_{K,\alpha}(j(x))p_0 + h_{B,\alpha}(j(x))(1 - p_0)}. \quad (5)$$

It can be shown that for  $\alpha > 0$  and under certain assumptions, the continuous analog of the measure  $S_\alpha(x)$  is stable with respect to small changes in the function  $F$  (more details in the on-line Appendix [B](#)). This indicates that the regularized measure is indeed quite robust. Another benefit of the regularization is that by suitable choices for  $\alpha$  it is possible to emphasize and de-emphasize different features that are used for the function  $F$ . Having a larger  $\alpha$  for a certain feature will decrease the weight of that feature in the resulting saliency map.

### 2.3 Implementation

For the feature function  $F$  we will use the CIELab color values of an image, and also motion information in the case of video sequences. For still images, if  $L(x)$ ,  $a(x)$ , and  $b(x)$  are the CIELab values at a point  $x$ , the feature map is  $F(x) = (L(x), a(x), b(x))$ . In the case of frames in a video sequence we combine the CIELab information for each frame with the magnitude of the optical flow  $Y(x)$ . The feature map is then  $F(x) = (L(x), a(x), b(x), Y(x))$ . All the values are quantized, and the bins are the elements in the finite feature space.

To simplify the computations, we make the assumption that the random variables  $L(Z)$ ,  $a(Z)$ ,  $b(Z)$ ,  $Y(Z)$  are independent in any subwindow. This is reasonable since the CIELab color space is constructed so that the intensity value  $L$  is independent of the  $a$  and  $b$  coordinates, and also since in our experiments using a joint distribution for  $a$  and  $b$  did not yield improved results compared to the case where independence was assumed. It is also fair to assume that the optical flow  $Y(Z)$  is independent of  $L(Z)$ ,  $a(Z)$ ,  $b(Z)$ .

Using the independence, we have  $P(F(x)|H_0) = \mathbf{h}_K(x)$  where  $\mathbf{h}_K(x)$  is the product of normalized histograms  $h_K^t(t(x))$  (here  $t$  is one of  $L$ ,  $a$ ,  $b$ ,  $Y$ ) and  $h_K^t(t_0)$  is equal to the number of points  $z$  in  $K$  such that  $t(z) = t_0$  divided by the number of points in  $K$ , etc. We define regularized histograms

$$h_{K,\alpha}^t(j) = \mathcal{N}\left(\sum_k g_\alpha(j - k)h_K^t(k)\right), \quad t \text{ is one of } L, a, b, Y. \quad (6)$$

Here  $\mathcal{N}(f(j)) = \frac{1}{\sum_k f(k)}f(j)$  is the normalization operator. The final saliency measure is given by

$$S_\alpha(x) = \frac{\mathbf{h}_{K,\alpha}(x)p_0}{\mathbf{h}_{K,\alpha}(x)p_0 + \mathbf{h}_{B,\alpha}(x)(1 - p_0)}. \quad (7)$$

<sup>1</sup> <http://www.ee.oulu.fi/mvg/page/saliency>

Here  $\mathbf{h}_{K,\alpha}(x)$  is equal to  $h_{K,\alpha}^L(L(x))h_{K,\alpha}^a(a(x))h_{K,\alpha}^b(b(x))$  for still images and to  $h_{K,\alpha}^L(L(x))h_{K,\alpha}^a(a(x))h_{K,\alpha}^b(b(x))h_{K,\alpha}^Y(Y(x))$  for frames in a video sequence, and  $\mathbf{h}_{B,\alpha}(x)$  is defined in a similar manner.

The saliency map for the entire image is achieved by sliding the window  $W$  with different scales over the image, constructing the proposed feature histograms for each window, smoothing the histograms, and then computing the measure for each pixel in  $K$  at each window position and scale. The final saliency value is then taken as the maximum over all windows containing a particular pixel. Figure 2 shows an illustration of the process.

In practice it is enough to evaluate the measure only in a small subset of all possible window positions and scales. In our experiments we used a regular grid with step size equal to 1 percent of the largest image dimension. We applied four scales with row and column sizes equal to  $\{25, 10; 30, 30; 50, 50; 70, 40\}$  percents of the largest image dimension, respectively. An illustrative Matlab implementation of our measure is available on-line<sup>2</sup>.

### 3 Salient Object Segmentation

In this section, we propose a bilayer segmentation method that estimates the salient and non-salient pixels of an image or a video by minimizing an energy function, which is derived from a conditional random field model that incorporates the pixelwise saliency measure of the previous section. The motivation for using a CRF model is the fact that usually the goal of saliency detection is to achieve an object-level segmentation rather than pixel-level segmentation. That is, the user is more interested in objects which contain salient pixels than the salient pixels themselves. Therefore, instead of considering pixels independently and segmenting the saliency maps by simple thresholding, it is reasonable to formulate the binary labeling problem in terms of a CRF based energy function, whose exact global minimum can be computed via graph cuts [27,28]. In the following, we describe the energy functions used in our experiments. The formulations are inspired by several previous works which apply graph cuts for binary segmentation problems, e.g. [29,30].

#### 3.1 Segmentation Energy for Still Images

First, given an image with  $N$  pixels, we use the saliency measure  $S_\alpha$  to compute a saliency map  $s = (s_1, \dots, s_N)$ , which is an array of saliency values. Further, we represent the image as an array  $c = (c_1, \dots, c_N)$ , where each  $c_n = (L_n, a_n, b_n)$  is a Lab color vector for a single pixel. Our task is to find a binary labeling  $\sigma = (\sigma_1, \dots, \sigma_N)$  so that  $\sigma_n \in \{0, 1\}$  indicates whether the pixel  $n$  belongs to a salient object or not.

<sup>2</sup> <http://www.ee.oulu.fi/mvg/page/saliency>



The optimal labeling is computed by minimizing the energy function

$$E_I(\sigma, c, s) = \sum_{n=1}^N (w_S U^S(\sigma_n, s_n) + w_C U^C(\sigma_n, c_n)) + \sum_{(n,m) \in \mathcal{E}} V(\sigma_n, \sigma_m, c_n, c_m), \tag{8}$$

which consists of two unary terms,  $U^S$  and  $U^C$ , and a pairwise term  $V$ , which depends on the labels of neighboring pixels<sup>3</sup>. The weight factors  $w_S$  and  $w_C$  are scalar parameters. The purpose of  $U^S$  is to penalize labelings which assign pixels with low  $s_n$  to the salient layer, whereas  $U^C$  encourages such labelings where the salient layer includes pixels which have similar colors as pixels for which  $s_n$  is high. The pairwise term  $V$  favors spatial continuity of labels. Overall, the energy function (8) has the standard form [28], which is used in many segmentation approaches [29] and can be statistically justified by using the well-known CRF formulation [30]. The precise definitions for  $U^S$ ,  $U^C$ , and  $V$  are described below.

The unary saliency term  $U^S$  is defined by

$$U^S(\sigma_n, s_n) = \delta_{\sigma_n,1}(1 - f(s_n)) + \delta_{\sigma_n,0}f(s_n), \tag{9}$$

where  $\delta_{\cdot,\cdot}$  is the Kronecker delta and  $f$  is defined by either

$$f(s_n) = \max(0, \text{sign}(s_n - \tau)) \quad \text{or} \quad f(s_n) = (s_n)^\kappa. \tag{10}$$

In probabilistic terms, one may think that  $U^S$  is an approximation to

$$-\log P(S_n = s_n | \sigma_n) = -\delta_{\sigma_n,1} \log p_1(s_n) - \delta_{\sigma_n,0} \log p_0(s_n), \tag{11}$$

where  $p_1$  and  $p_0$  are the conditional density functions of  $s_n$  given that pixel  $n$  is salient or non-salient, respectively. Hence, loosely speaking, the ratio  $f(s_n) : (1 - f(s_n))$  can be seen as a one-parameter model for the ratio of negative log-likelihoods,  $(-\log p_0(s_n)) : (-\log p_1(s_n))$ .

The unary color term  $U^C$  is defined by

$$U^C(\sigma_n, c_n) = -\log P(C_n = c_n | \sigma_n) = -\delta_{\sigma_n,1} \log p_1^c(c_n) - \delta_{\sigma_n,0} \log p_0^c(c_n), \tag{12}$$

where the conditional density functions  $p_1^c$  and  $p_0^c$  are the color distributions of salient and non-salient pixels, respectively. Given image  $c$ , we compute  $p_1^c$  and  $p_0^c$  as a product of two histograms, that is,  $p_1^c(c_n) = h_1^L(L_n)h_1^{ab}(a_n, b_n)$  and  $p_0^c(c_n) = h_0^L(L_n)h_0^{ab}(a_n, b_n)$ . The histograms  $h_1^L$  and  $h_0^L$  are computed as weighted histograms of pixels' intensity values, where the weights for pixel  $n$  are  $f(s_n)$  and  $(1 - f(s_n))$ , respectively. The color histograms  $h_1^{ab}$  and  $h_0^{ab}$  are computed in a similar manner using  $f(s_n)$  and  $(1 - f(s_n))$  as weights.

The pairwise prior  $V$  is

$$V(\sigma_n, \sigma_m, c_n, c_m) = \gamma \delta_{\sigma_n, \sigma_m} e^{-\|c_n - c_m\|_\Lambda^2} + \eta \delta_{\sigma_n, \sigma_m}, \tag{13}$$

where  $\gamma$  and  $\eta$  are scalar parameters and  $\|\cdot\|_\Lambda$  is a Mahalanobis distance with diagonal matrix  $\Lambda$ . Both terms in (13) penalize neighboring pairs of pixels that

<sup>3</sup> Set  $\mathcal{E}$  contains pairs  $(n, m)$  for which  $n < m$  and pixels  $n$  and  $m$  are 4-connected.

have different labels. However, the first term adds lower cost for such segmentation boundaries that co-occur with contours of high image contrast [30].

Given image  $c$  and saliency map  $s$ , we estimate  $\sigma$  by minimizing (8) via graph cuts. The pixels labeled by 1 belong to salient objects and the rest is background. Further, given test images with ground truth saliency maps where the pixels of salient objects are manually labeled, we compare the two choices for  $f$  in (10) by computing the corresponding ROC curves. That is, by changing the value of parameter  $\tau$  (or  $\kappa$ ) from 0 to  $\infty$  the labeling gradually changes from one map to zero map, and we may draw a ROC curve by counting the number of correctly and incorrectly labeled pixels at each parameter setting. Section 4 reports the results obtained with a publicly available dataset of 1000 images. For the experiments, we determined the values of  $w_S$ ,  $w_C$ ,  $\gamma$ , and  $\eta$  by the approach in [31]. All other parameters except  $\tau$  and  $\kappa$  were set to manually predefined values and kept constant during the experiments.

### 3.2 Segmentation Energy for Videos

Our CRF segmentation model for videos incorporates motion information indirectly via the saliency measure  $S_\alpha$ , as described in Section 2, but also directly via an additional unary term, which is introduced below. In detail, the energy function for videos is an augmented version of (8), i.e.

$$E_V(\sigma^t, \sigma^{t-1}, \sigma^{t-2}, c^t, c^{t-1}, s) = E_I(\sigma^t, c^t, s) + \sum_{n=1}^N U^T(\sigma_n^t, \sigma_n^{t-1}, \sigma_n^{t-2}, c_n^t, c_n^{t-1}) \quad (14)$$

where  $\sigma^t$  is the segmentation of the current frame,  $\sigma^{t-1}$  and  $\sigma^{t-2}$  are the segmentations of the two previous frames,  $c^t$  is the current frame,  $c^{t-1}$  is the previous frame, and  $U^T$  is an additional unary term which improves temporal coherence.

The term  $U^T$  has the following form,

$$U^T(\sigma_n^t, \sigma_n^{t-1}, \sigma_n^{t-2}, c_n^t, c_n^{t-1}) = \mu \delta_{\sigma_n^t, \sigma_n^{t-1}} e^{-\|c_n^t - c_n^{t-1}\|_\Gamma^2} - \nu \log p_T(\sigma_n^t | \sigma_n^{t-1}, \sigma_n^{t-2}), \quad (15)$$

where  $\mu$  and  $\nu$  are scalar parameters,  $\|\cdot\|_\Gamma$  is a Mahalanobis distance with diagonal matrix  $\Gamma$ , and  $p_T$  is the prior probability density function of  $\sigma_n^t$  conditioned on  $\sigma_n^{t-1}$  and  $\sigma_n^{t-2}$ . Thus, since  $p_T(\sigma_n^t = 0 | \sigma_n^{t-1}, \sigma_n^{t-2}) = 1 - p_T(\sigma_n^t = 1 | \sigma_n^{t-1}, \sigma_n^{t-2})$ ,  $p_T$  is defined by four parameters which determine  $p_T(\sigma_n^t = 1 | \sigma_n^{t-1}, \sigma_n^{t-2})$  corresponding to the following four cases:  $(\sigma_n^{t-1}, \sigma_n^{t-2}) = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$ . The first term in (15) is an additional data-dependent cost for pixels which change their label between frames  $(t-1)$  and  $t$ . This extra cost is smaller for those pixels whose color changes a lot between the frames.

Given a video sequence, we compute the segmentation  $\sigma^t$  for frames  $t > 2$  by minimizing (14) via graph cuts. For grayscale videos we use the grayscale version of (14). In the experiments, the values of the common parameters were the same for the grayscale and color versions. Further, we used the first choice for  $f$  in (10) and all parameter values were kept constant in the experiments.

## 4 Experiments

In this section, we assess the proposed approach in saliency segmentation experiments. The performance is compared with the state-of-the-art methods using the programs given by the authors [10,23,21,20,26] or our own implementation with default parameters [24]. The experiments are divided into two parts, where the first one considers still images and the second one video sequences.

### 4.1 Segmenting Salient Objects from Images

First, we run the publicly available saliency segmentation test, introduced in [26]. The proposed method is compared to the band-pass approach in [26], which was reported to achieve clearly the best performance among the several tested methods [26] (note the erratum<sup>4</sup>). In addition we also include the approaches from [21] and [24], since they were not evaluated in [26].

The experiment contains 1000 color images with pixel-wise ground truth segmentations provided by human observers. First a saliency map is computed for each test image and then a segmentation is generated by simply thresholding the map by assigning the pixels above the given threshold as salient (white foreground) and below the threshold as non-salient (black background). A precision and recall rate is then computed using definitions:

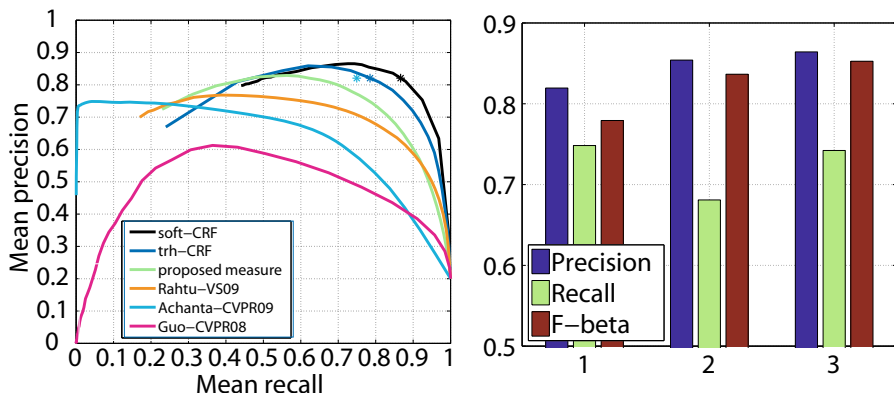
$$precision = |SF \cap GF|/|SF|, \quad recall = |SF \cap GF|/|GF|, \quad (16)$$

where  $SF$  denotes the segmented foreground pixels,  $GF$  denotes the ground truth foreground pixels, and  $|\cdot|$  refers to number of elements in a set. By sliding the threshold from minimum to maximum saliency value, we achieved the precision-recall curves illustrated in Figure 3 (magenta, cyan, orange, and green).

The results show that the proposed saliency measure achieves the highest performance up to a recall rate 0.9. Furthermore also the method from [21] seems to outperform the state-of-the-art results in [26]. Notice that the precision-recall curves of the proposed method and the method in [21] do not have values for small recalls because several pixels reach the maximum saliency value and they change labels simultaneously when the threshold is lowered below one. At maximum recall all methods converge to 0.2 precision, which corresponds to a situation where all pixels are labeled as foreground.

We continue the experiment by adding the CRF segmentation model from Section 3 on top of our saliency measure. First, we perform the same experiment as above, but refine the thresholded saliency maps using the CRF model (i.e. the first choice is used for  $f$  in (10)). The resulting precision-recall-curve in Figure 3 (blue) illustrates a clear gain compared to thresholded saliency map in both precision and recall. Finally, we replace the thresholded saliency maps in the CRF by the soft assignment approach of Section 3 (i.e. the second choice for  $f$  in (10)). Now, instead of sliding threshold  $\tau$  we change the exponent  $\kappa$ , and

<sup>4</sup> [http://ivrg.epfl.ch/supplementary\\_material/RK\\_CVPR09/index.html](http://ivrg.epfl.ch/supplementary_material/RK_CVPR09/index.html)



**Fig. 3.** Left: Mean precision-recall curves using comparison methods and the proposed approach. Right: Mean precision, recall, and F-measure values for comparison method [26] (1), our method with thresholding (2), and our method with soft assignments (3). Notice that  $\beta = 0.3$  (used according to [26]) strongly emphasizes precision.

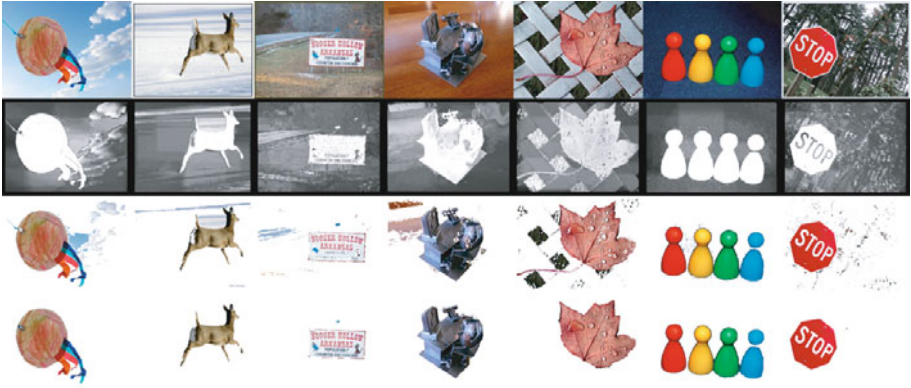
achieve the corresponding precision-recall-curve in Figure 3 (black), which shows further improvement in performance.

In [26] the best results were achieved by combining the band-pass saliency map with adaptive thresholding and the mean-shift segmentation algorithm. The achieved precision, recall, and F-measure values were 0.82, 0.75, and 0.78, respectively. The F-measure was computed from precision and recall by  $F_\beta = (1 + \beta^2) (\textit{precision} \cdot \textit{recall}) / (\beta^2 \cdot \textit{precision} + \textit{recall})$ , where  $\beta = 0.3$  was used [26]. This corresponds to a point marked using by a cyan star in Figure 3. This result remains lower than our results with both naïve thresholding and soft mapping with CRF, which provide the same precision with recalls 0.79 and 0.87, respectively. These points are also marked in Figure 3 with correspondingly colored stars. The maximum F-measure value we achieve is 0.85, which represents 9 percent improvement over [26]. The comparison of F-measures is shown in Figure 3. A few results of the proposed saliency segmentation method are shown in Figure 4 for subjective evaluation.

## 4.2 Segmenting Salient Objects from Video Sequences

Another set of experiments was performed using videos. The saliency maps were computed as described in Section 2 by using both the CIE Lab color values (only L in the case of gray-scale videos) and the magnitude of optical flow as features. The optical flow was computed using a publicly available<sup>5</sup> implementation [32], which can provide real-time performance. The final salient segments were computed using either direct thresholding or the CRF method of Section 3.

<sup>5</sup> <http://gpu4vision.org/>



**Fig. 4.** Examples of saliency maps and segmentations. Top row shows the original image, second row shows the saliency maps, third row shows the segmentations using threshold 0.7, and bottom row shows the segmentations using the CRF model.

The results are compared with methods in [24,21,20,10] from which the last mentioned is a general background subtraction method. All comparison methods used default parameters given by the authors. Further, in order to achieve best possible performance with comparison methods, we also included all the post-processing techniques presented in the original papers. As test videos, we used the publicly available image sequences originally used in [21] and [22]. The two sequences from [21] illustrate moving and stationary objects in the case of a fixed and a mobile camera. Sequences from [22] show highly dynamic backgrounds with targets of various size. The original results of [22] are available on-line and are directly comparable to our results. Their experiments also include several traditional background subtraction approaches.

Figure 5 illustrates characteristic frames from tested sequences. The results include original frames, saliency maps, and final segmentations. Full videos are also available on-line<sup>6</sup>. The results illustrate the problems of traditional background subtraction methods, which work well with stationary cameras and constantly moving objects. However serious problems appear if the camera is moving and targets may stop every once in a while. The poor resolution of [24] is visible in the inaccurate segmentations and several missed objects. The method in [21] works better, but the result is rather noisy and the segmentations are not accurate. The missing motion information is also visible in the results with [21].

The proposed approach achieves the most stable results, where also the effect of motion cues is clearly visible. The returned segments mostly correspond to natural objects. Sometimes the method may return salient segments which a human observer would classify as part of the background (e.g. grass between the roads). However, like with all saliency detection methods this is difficult to avoid if these objects are distinct from the background in terms of visual contrast.

<sup>6</sup> <http://www.ee.oulu.fi/mvg/page/saliency>



**Fig. 5.** Results of saliency detection from videos. Each group of eight images correspond to one test sequence and they are organized as follows: Left column from top to bottom consists of the original frame, the proposed saliency map, segmentation by thresholding proposed saliency map, and segmentation using the proposed CRF model. Right column from top to bottom consists of segmentations using the saliency maps and full post processing of comparison methods [10], [24], [20], and [21], respectively.

## 5 Conclusions

In this paper, we presented a new combination of a saliency measure and a CRF based segmentation model. The measure was formulated using a probabilistic framework, where different features were fused together in joint distributions. The sensitivity of the proposed measure was shown to be controlled by a smoothing parameter, which can also be used to set the relative weights of the features.

The resulting saliency map was turned into a segmentation of natural and well-defined objects using the CRF model. The segmentations were constantly improved and stabilized especially in the case of video sequences, where the smoothness over frames was emphasized by the applied model. In addition we proposed a technique to include optical flow motion cues into the saliency estimation, which greatly improved the recall rate with videos.

The experiments with a publicly available dataset showed that our approach yields clearly higher performance than the state-of-the-art in terms of both recall and precision. The new method produces both more discriminative saliency maps and more accurate segmentations. The precision was improved especially at high recalls, where previous results were rather poor. The experiments with video sequences showed also consistent improvement over the tested methods.

The features used in our approach included Lab color values and optical flow, which are both obtainable in real-time. The saliency measure itself was evaluated using sliding windows and integral histograms. The processing takes about 8 seconds per image with our current Matlab implementation, but we believe that this can be reduced to close to real time. The CRF energy minimization by graph cuts took 1/20 seconds per image. In future, we aim to achieve a real time implementation by using total-variation techniques instead of graph cuts.

**Acknowledgments.** The work has been supported by the Academy of Finland.

## References

1. Yarbus, A.: Eye movements and vision. Plenum, New York (1967)
2. Avidan, S., Shamir, A.: Seam carving for content-aware image resizing. *ACM Transactions of Graphics* 26 (2007)
3. Marchesotti, L., Cifarelli, C., Csurka, G.: A framework for visual saliency detection with applications to image thumbnailing. In: *ICCV* (2009)
4. Han, J., Ngan, K.N., Li, M., Zhang, H.: Unsupervised extraction of visual attention objects in color images. *IEEE Trans. Circuits Syst. Video Techn.* 16, 141–145 (2006)
5. Ko, B., Nam, J.: Object-of-interest image segmentation based on human attention and semantic region clustering. *J. Opt. Soc. Am.* 23, 2462–2470 (2006)
6. Rutishauser, U., Walther, D., Koch, C., Perona, P.: Is bottom-up attention useful for object recognition. In: *CVPR* (2004)
7. Yang, L., Zheng, N., Yang, J., Chen, M., Cheng, H.: A biased sampling strategy for object categorization. In: *ICCV* (2009)
8. Sheikh, Y., Shah, M.: Bayesian modeling of dynamic scenes for object detection. *IEEE TPAMI* 27, 1778–1792 (2005)
9. Monnet, A., Mittal, A., Paragios, N., Ramesh, V.: Background modeling and subtraction of dynamic scenes. In: *CVPR*, vol. 2, pp. 1305–1312 (2003)

10. Heikkilä, M., Pietikäinen, M.: A texture-based method for modeling the background and detecting moving objects. *IEEE TPAMI* 28, 657–662 (2006)
11. Ren, Y., Chua, C., Ho, Y.: Motion detection with nonstationary background. *Machine Vision and Applications* 13, 332–343 (2003)
12. Sheikh, Y., Javed, O., Kanade, T.: Background Subtraction for Freely Moving Cameras. In: *ICCV* (2009)
13. Lampert, C., Blaschko, M., Hofmann, T.: Efficient subwindow search: A branch and bound framework for object localization. *IEEE TPAMI* 31, 2129–2142 (2009)
14. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for multi-class object layout. In: *ICCV* (2009)
15. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *CVPR*, vol. 1 (2005)
16. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE TPAMI* 20, 1254–1259 (1998)
17. Itti, L., Koch, C.: A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research* 40, 1489–1506 (2000)
18. Ma, Y., Zhang, H.: Contrast-based image attention analysis by using fuzzy growing. In: *ACM Intl. Conf. on Multimedia*, pp. 59–68 (2003)
19. Hu, Y., Xie, X., Ma, W., Chia, L., Rajan, D.: Salient region detection using weighted feature maps based on the human visual attention model. In: Aizawa, K., Nakamura, Y., Satoh, S. (eds.) *PCM 2004*. LNCS, vol. 3332, pp. 993–1000. Springer, Heidelberg (2004)
20. Achanta, R., Estrada, F.J., Wils, P., Süsstrunk, S.: Salient region detection and segmentation. In: Gasteratos, A., Vincze, M., Tsotsos, J.K. (eds.) *ICVS 2008*. LNCS, vol. 5008, pp. 66–75. Springer, Heidelberg (2008)
21. Rahtu, E., Heikkilä, J.: A simple and efficient saliency detector for background subtraction. In: *IEEE Intl. Workshop on Visual Surveillance*, pp. 1137–1144 (2009)
22. Mahadevan, V., Vasconcelos, N.: Spatiotemporal saliency in highly dynamic scenes. *IEEE TPAMI* 32, 171–177 (2010)
23. Hou, X., Zhang, L.: Saliency detection: A spectral residual approach. In: *CVPR* (2007)
24. Guo, C., Ma, Q., Zhang, L.: Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In: *CVPR* (2008)
25. Gao, D., Vasconcelos, N.: Bottom-up saliency is a discriminant process. In: *ICCV* (2007)
26. Achanta, R., Hemami, S.S., Estrada, F.J., Süsstrunk, S.: Frequency-tuned salient region detection. In: *CVPR* (2009)
27. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE TPAMI* 23, 1222–1239 (2001)
28. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? *IEEE TPAMI* 26, 147–159 (2004)
29. Rother, C., Kolmogorov, V., Blake, A.: "Grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.* 23, 309–314 (2004)
30. Criminisi, A., Cross, G., Blake, A., Kolmogorov, V.: Bilayer segmentation of live video. In: *CVPR*, vol. 1, pp. 53–60 (2006)
31. Szummer, M., Kohli, P., Hoiem, D.: Learning CRFs using graph cuts. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part II*. LNCS, vol. 5303, pp. 582–595. Springer, Heidelberg (2008)
32. Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., Bischof, H.: Anisotropic Huber-L1 optical flow. In: *BMVC* (2009)



# ClassCut for Unsupervised Class Segmentation

Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari

Computer Vision Laboratory, ETH Zurich, Zurich, Switzerland  
{bogdan,deselaers,ferrari}@vision.ee.ethz.ch

**Abstract.** We propose a novel method for unsupervised class segmentation on a set of images. It alternates between segmenting object instances and learning a class model. The method is based on a segmentation energy defined over all images at the same time, which can be optimized efficiently by techniques used before in interactive segmentation. Over iterations, our method progressively learns a class model by integrating observations over all images. In addition to appearance, this model captures the location and shape of the class with respect to an automatically determined coordinate frame common across images. This frame allows us to build stronger shape and location models, similar to those used in object class detection. Our method is inspired by interactive segmentation methods [1], but it is fully automatic and learns models characteristic for the object class rather than specific to one particular object/image. We experimentally demonstrate on the Caltech4, Caltech101, and Weizmann horses datasets that our method (a) transfers class knowledge across images and this improves results compared to segmenting every image independently; (b) outperforms Grabcut [1] for the task of unsupervised segmentation; (c) offers competitive performance compared to the state-of-the-art in unsupervised segmentation and in particular it outperforms the topic model [2].

## 1 Introduction

Image segmentation is a fundamental problem in computer vision. Over the past years methods that use graph-cut to minimize binary pairwise energy functions have become the de-facto standard for segmenting specific objects in individual images [1, 3, 4]. These methods employ appearance models for the foreground and background which are estimated through user interactions [1, 3, 4].

On the one hand, analog approaches have been presented for object class segmentation where the appearance models are learned from a set of training images with ground-truth segmentations [5–7]. However, obtaining ground-truth segmentations is cumbersome and error-prone.

On the other hand, approaches to unsupervised class segmentation have also been proposed [2, 8–10, 12, 13]. In unsupervised segmentation a set of images depicting different instances of an object class is given, but without information about the appearance and shape of the objects to be segmented. The aim of an algorithm is to automatically segment the object instance in each image.

Interestingly, most previous approaches to unsupervised segmentation do not use energy functions similar to those in interactive and supervised segmentation, but instead use topic models [2] or other specialized generative models [10, 12] to find recurring patterns in the images.

We propose ClassCut, a novel method for unsupervised segmentation based on a binary pairwise energy function similar to those used in interactive/supervised segmentation. As opposed to those, our energy function is defined over a set of images rather than on one image [1, 3–5]. Inspired by GrabCut [1], where the two stages of learning the foreground/background appearance models and segmenting the image are alternated, our method alternates between learning a *class model* and segmenting the objects in all images jointly. The class model is learned from all images at the same time, so as to capture knowledge about the class rather than specific to one image [1]. Therefore, it helps the next segmentation iteration, as it transfers between images knowledge about the appearance and shape of the class. Thanks to the nature of our energy function, we can segment all images jointly using existing efficient algorithms used in interactive segmentation approaches [1, 3, 14, 15].

Inspired by representations successfully used in supervised object class detection [16, 17], our approach anchors the object class in a reference coordinate frame common across images. This enables modeling the spatial structure and shape of the class, as well as designing novel priors tailored to the unsupervised segmentation task. We determine this reference frame automatically in every image with a procedure based on a salient object detector [18].

At each iteration ClassCut updates the class model, which captures the appearance, shape, and location distribution of the class within the reference frame. The final output of the method are images with segmented object instances as well as the class model.

In the experiments, we demonstrate that our method (a) transfers knowledge between images and this improves the performance over segmenting each image independently; (b) outperforms the original GrabCut [1], which is the main inspiration behind it and turns out to be a very competitive baseline for unsupervised segmentation; (c) offers competitive performance compared to the state-of-the-art in unsupervised segmentation; (d) learns meaningful, intuitive class models. Source code for ClassCut is available at <http://www.vision.ee.ethz.ch/~calvin>.

**Related Work.** We discussed in the introduction that our method employs energy minimization techniques used in interactive segmentation [1, 3, 4, 14, 15], and how it is related to supervised [5, 7, 19] as well as to unsupervised [2, 10–12] class segmentation methods.

A different task is object discovery, which aims at finding multiple object classes from a mixed set of unlabeled images [11, 29]. In our work instead, all images contain instances of one class.

The two closest work to ours are [8, 9], which have a procedure iterating between updating a model and segmenting the images. In [8] the model is given a set of class and non-class images and then it iteratively improves the foreground/background labeling of image fragments based on their class likelihoods.

Their method learns local segmentations masks for image fragments, while our method learns a more complete class model, including appearance, shape and location in a global reference frame.

Arora et al. [9] learn a template consistent over all images using variational inference. Their template model is very different from our class model, and closer to a constellation model [20]. Moreover, their method optimizes the segmentation of the images individually rather than jointly.

Finally, our approach is also related to co-segmentation [21] where the goal is to segment a specific object from two images at the same time. Here we try to go a step further and co-segment a set of images showing different object instances of an unknown class.

## 2 Overview of Our Method

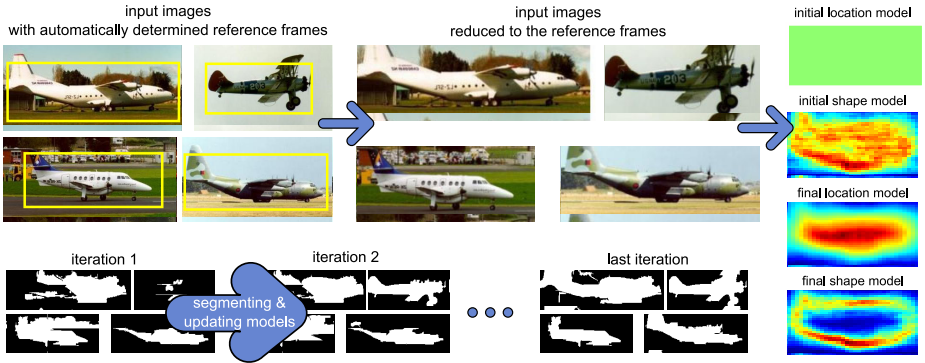
The goal is to jointly segment objects of an unknown class from a set of images. Analog to the scheme of GrabCut [1], ClassCut alternates two stages: (1) learning/updating a class model given the current segmentations (sec. 4); (2) jointly segmenting the objects in all images given the current class model (sec. 3). It converges when the segmentation is unchanged in two consecutive iterations.

Our segmentation model for stage (2) is a binary pairwise energy function, which can be optimized efficiently by techniques used in interactive segmentation [1, 3, 22], but jointly over all images rather than on a single image [1] (sec. 3).

In stage (1), learning the class model over all images at once enables capturing knowledge characteristic for the *class* rather than specific to a particular *image* [1]. As the class model is used in the next segmentation iteration it transfers knowledge across images, typically from easier images to more difficult ones, aiding their segmentation. For example, the model might learn in the first iteration that airplanes are typically grayish and the background is often blue (fig. 1). In the next iteration, this will help in images where the airplane is difficult to segment (e.g. because of low contrast).

The class model we propose (sec. 3.2) consists of several components modeling different class characteristics: appearance, location, and shape. In addition to a color component also used in GrabCut [1], the appearance model includes a bag-of-words [23] of SURF descriptors [24], which is well suited for modeling class appearance. Moreover, we model the location (sec. 3.2) and shape (sec. 3.2) of the object class w.r.t. a reference coordinate frame common across images (sec. 5). Overall, our model focuses on knowledge at the class level rather than at the level of one object as in the works it is inspired from [1, 4].

In addition to the class model, the segmentation energy include priors tailored for segmenting classes (sec. 3.1). The priors are defined on superpixels [25], which act as grouping units for homogeneous areas. Superpixels bring two advantages: (i) they provide additional structure, i.e. the set of possible segmentations is reduced to those aligning well with image boundaries; (ii) they reduce the computational complexity of segmentation. We formulate four class segmentation priors over superpixels and multiple images (sec. 3.1).



**Fig. 1. Overview of our method.** The top row shows the input images, the automatically determine reference frames and the initial location and shape models. The bottom row shows how the segmentations evolve over the iterations as well as the final location and shape models.

If a common reference frame on the objects is available, our method exploits it to anchor the location and shape models to it and to improve the effectiveness of some of the priors. We apply a salient object detector [18] to determine this reference frame automatically (sec. 5). In sec. 6 we show how this detector improves segmentation results compared to using the whole image as a reference frame. Fig. 1 shows an overview of the entire method.

### 3 Segmentation

In the set of images  $\mathcal{I} = \{I_1, \dots, I_N\}$  each image  $I_n$  (given either as a full image or as automatically determined reference frame) consists of superpixels  $\{S_n^1, \dots, S_n^{K_n}\}$ . We search for the labeling  $L^* = ((l_1^1, \dots, l_1^{K_1}), \dots, (l_n^1, \dots, l_n^{K_n}), \dots, (l_N^1, \dots, l_N^{K_N}))$  that sets  $l_n^k = 1$  for all superpixels  $S_n^k$  on the foreground and  $l_n^j = 0$  for all superpixels  $S_n^j$  on the background.

To determine  $L^*$ , we minimize

$$L^* = \arg \min_L \{E_\Theta(L, \mathcal{I})\} \quad \text{with} \quad E_\Theta(L, \mathcal{I}) = \Phi_\Theta(L, \mathcal{I}) + \Psi_\Theta(L, \mathcal{I}) \quad (1)$$

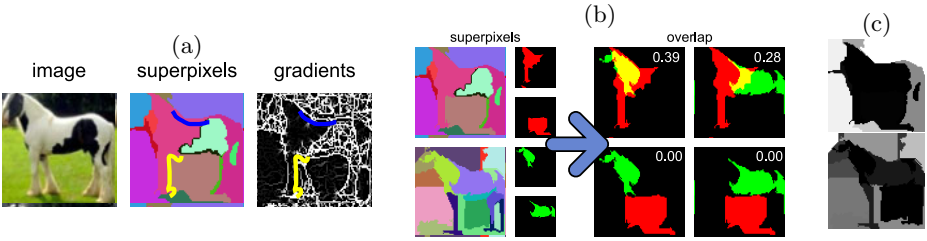
where  $\Phi$  is the segmentation prior (sec. 3.1) and  $\Psi$  is the class model (sec. 3.2). In sec. 3.3 we describe how to minimize eq. (1).  $\Theta$  are the parameters of the model.

#### 3.1 Prior $\Phi_\Theta(L, \mathcal{I})$

The prior  $\Phi$  consists of four terms

$$\Phi_\Theta(L, \mathcal{I}) = w_\Lambda \Lambda(L, \mathcal{I}) + w_\chi \chi(L, \mathcal{I}) + w_\Gamma \Gamma(L, \mathcal{I}) + w_\Delta \Delta(L, \mathcal{I}) \quad (2)$$

The scalars  $w$  are part of the model parameters  $\Theta$  and weight the terms. Below we describe the terms in detail.



**Fig. 2. Priors.** (a) The smoothness prior between two superpixels is weighted inversely to the sum over the gradients along their boundary (shown in yellow and blue for two pairs of superpixels). (b) The between image smoothness prior is weighted by the overlap (yellow) of superpixels (shown for two pairs of superpixels (red/green) in two images). (c) The border penalty assigns high values to superpixels touching the reference frame boundary (dark=low values, bright=high values).

**The Within Image Smoothness  $\Lambda$**  is a smoothness prior for superpixels which generalizes the pixel-based smoothness priors typically used in interactive segmentation [1]. It penalizes neighboring superpixels having different labels.

$$\Lambda(L, \mathcal{I}) = \sum_n \sum_{j,k} \delta(l_n^j \neq l_n^k) \exp(-\text{grad}(S_n^j, S_n^k)) \quad (3)$$

where  $j, k$  are the indices of neighboring superpixels  $S_n^j, S_n^k$  within image  $I_n$ .  $\delta(l_n^j \neq l_n^k) = 1$  if the labels  $l_n^j, l_n^k$  are different and 0 otherwise. The gradient  $\text{grad}(S_n^j, S_n^k)$  between  $S_n^j$  and  $S_n^k$  is computed by summing the gradient magnitudes [26] along the boundary between  $S_n^j, S_n^k$  (fig. 2a) normalized w.r.t. the length of the boundary. Thus, the penalty is smaller if the two superpixels are separated by high gradients. This term encourages segmentations aligned with the image gradients.

**The Between Image Smoothness  $\chi$**  operates on superpixels across images. It encourages superpixels in different images but with similar location w.r.t. the reference frame to have the same label:

$$\chi(L, \mathcal{I}) = \sum_{n,m} \sum_{j,k} \delta(l_n^j \neq l_m^k) \frac{|S_n^j \cap S_m^k|}{|S_n^j \cup S_m^k|} \quad (4)$$

where  $n, m$  are two images and  $j, k$  superpixels, one in  $I_n$ , the other in  $I_m$ . This penalty grows with the overlap of the superpixels (measured as area of intersection over area of union). Therefore only overlapping superpixels interact (fig. 2b). This term encourages similar segmentations across all images (w.r.t. the reference frame).

**The Border Penalty  $\Gamma$**  prefers superpixels at the image boundary to be labeled background. Objects rarely touch the boundary of the reference frame. Notice how the object would touch even a tight bounding-box around itself only in a few points (e.g. fig. 2a). The border penalty

$$\Gamma(L, \mathcal{I}) = \sum_n \sum_k l_n^k \frac{\text{border}(S_n^k)}{\text{perimeter}(S_n^k)} \quad (5)$$

assigns a penalty proportional to the number of pixels touching the reference frame ( $\text{border}(S_n^k)$ ) to each superpixel  $S_n^k$  normalized by its perimeter ( $\text{perimeter}(S_n^k)$ ). This term penalizes superpixels touching the border of the reference frame to be labeled foreground (fig. 2).

$\Gamma$  is only meaningful on superpixels. If the segmentation is performed at the pixel-level, the border penalty can be compared to a low prior on the boundary pixels which may be propagated toward the image center using the smoothness prior. This shows how superpixels introduce additional structure into the model.

**The Area Reward  $\Delta$**  encourages a large foreground region in order to find the entire recurring object and not just a small recurring object part. The term

$$\Delta(L, \mathcal{I}) = \sum_n \sum_m -l_n^m \frac{|S_n^m|}{|I_n|} \quad (6)$$

assigns to each superpixel a reward proportional to its area (normalized w.r.t. the area of the reference frame).

The combined effects of  $\Gamma$  and  $\Delta$  are similar to the (more complex) bounding-box prior [4]: the foreground region should be as large as possible while not crossing the boundary of the reference frame (here, touching it).

### 3.2 Class Model $\Psi_\Theta(L, \mathcal{I})$

The class model  $\Psi_\Theta(L, \mathcal{I})$  accounts for the appearance, shape, and location of the objects:

$$\Psi_\Theta(L, \mathcal{I}) = w_\Omega \Omega_\Theta(L, \mathcal{I}) + w_\Pi \Pi_\Theta(L, \mathcal{I}) + \sum_f w_{\Upsilon^f} \Upsilon_\Theta^f(L, \mathcal{I}) \quad (7)$$

The scalars  $w$  are part of the model parameters  $\Theta$  and weight the terms. Below we describe these models in detail. In sec. 4 we explain how they are initialized and updated over the iterations.

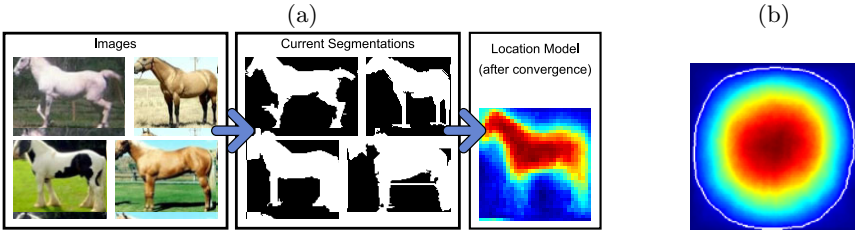
**The Location Model  $\Omega$**  accounts for the locations of objects w.r.t. the reference frames. We model the probability for a pixel  $s$  at its position to be foreground  $p^\Omega(l|s)$  as the empirical probability in the reference frame.  $p^\Omega$  is quantized to  $32 \times 32$  locations within the reference frame.

To compute the energy contribution for a superpixel  $S_n^k$  labeled foreground, we average over all positions in  $S_n^k$  and incorporate this into eq. (7) as

$$\Omega_\Theta(L, \mathcal{I}) = \sum_n \sum_k \frac{1}{|S_n^k|} \sum_{s \in S_n^k} -\log p^\Omega(l_n^k | s) \quad (8)$$

Fig. 3a shows a final location model obtained after convergence. The location model encourages similar segmentations w.r.t. the reference frame in all images.

**The Shape Model  $\Pi$**  accounts for the global shape of the objects within the reference frames. We model the global shape of the objects as the probability



**Fig. 3.** (a) **Training the location model  $\Omega$ .** In each iteration, we segment all images and reestimate a location model specific to the current class using the current segmentations. (b) **Generic object location prior.** The initial segmentation used to initialize appearance models is drawn in white.

$p^\Pi(\text{boundary}|s, \beta)$  that an object boundary with orientation  $\beta$  is at position  $s$ . This is modeled as the empirical probability of oriented object boundaries quantized into 5 orientations and  $32 \times 32$  spatial bins.

For a pair of neighboring superpixels  $S_n^j, S_n^k$  in image  $I_n$  this probability is accumulated along their boundary  $S_n^j \cap S_n^k$  to obtain the probability that one of them is foreground and the other background as:

$$p^\Pi(l_n^j \neq l_n^k | S_n^j, S_n^k) = \frac{1}{|S_n^j \cap S_n^k|} \sum_{s \in S_n^j \cap S_n^k} p^\Pi(\text{boundary}|s, \beta_s) \quad (9)$$

where  $\beta_s$  is orientation of pixel  $s$ . This model is then incorporated in eq. (7) as:

$$\Pi_\Theta(L, \mathcal{I}) = \sum_n \sum_{j,k} \delta(l_n^j \neq l_n^k) (\mu - p^\Pi(l_n^j \neq l_n^k | S_n^j, S_n^k)) \quad (10)$$

where  $\mu = \frac{1}{5 \cdot 32 \cdot 32} \sum_{s, \beta} p^\Pi(\text{boundary}|s, \beta)$  is the mean probability of a boundary over all locations and orientations.

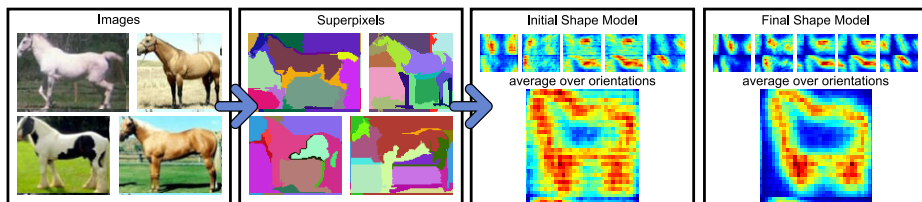
Fig. 4 shows an initial shape model and a shape model after convergence. The shape model encourages segmentations with similar shapes w.r.t. the reference frame in all images.

**The Appearance Models  $\Upsilon^f$**  capture the visual appearance of the foreground and background regions according to different visual descriptors  $f$ . As visual descriptors  $f$  we use color distributions (COL) and bag-of-words [23] of SURF descriptors [24] (BOW).

For a pixel  $s$ , the probability to be foreground (or background)  $p^f(l|s)$  is modelled using Gaussian mixtures for  $p^{\text{COL}}(l|s)$ , closely following [1], and using empirical probabilities for  $p^{\text{BOW}}(l|s)$ . It is incorporated into eq. (11) by averaging over all pixels within a superpixel.

Note that our appearance model extends the model of GrabCut [1] by the bag of SURF descriptor which is known to perform well for object classes.

$$\Upsilon_\Theta^f(L, \mathcal{I}) = \sum_n \sum_k -\frac{1}{|S_n^k|} \sum_{s \in S_n^k} \log p^f(l_n^k | s) \quad (11)$$



**Fig. 4. The shape model.** We initialize our shape model  $\Pi$  using only boundaries between superpixels. The shape model after convergence is shown on the right.

The appearance models capture the appearance of foreground and background region. The color model closely resembles those used in interactive segmentation and together with the bag-of-SURF model captures class appearance.

### 3.3 Energy Minimization

As the energy (eq. (11)) is defined over binary variables and comprises only unary  $(I, \Delta, \Omega, \mathcal{Y}_f)$  and pairwise  $(\chi, A, \Pi)$  terms, we minimize it using QPBO [22]. Since QPBO labels only those superpixels for which it is guaranteed to have the global optimum, some superpixels might be left unlabeled. To label these superpixels we use TRW-S [15]. TRW-S not only labels them but also computes a lower bound on the energy which may be used to assess how far from the global optimum the solution is.

Note that all pairwise terms except for the shape model are submodular. We observed that on average only about 2% of the pairwise terms in the final model (i.e. incorporating all cues) are non-submodular.

In our experiments, we observed that QPBO labels on average 91% of the superpixels according to the global optimum.

Furthermore, we observed that the minimization problem is hardest in the first few iterations and easier in the later iterations: over the iterations QPBO labels more superpixels and the difference between the lower bound and the actual energy of the solutions is also decreased.

## 4 Initializing and Updating the Class Model

We describe how to initialize the model and how to update the parameters of the class models at each iteration.

### 4.1 Location Model

The location model  $\Omega$  is initialized uniformly. At each iteration, we update the parameters of the location model using the current segmentation of all images of the current class according to the maximum likelihood criterion (fig. 3a): for each cell in the  $32 \times 32$  grid we reestimate the empirical probability of foreground using the current segmentations.



## 4.2 Shape Model

The shape model  $\mathcal{I}$  is initialized by accumulating the boundaries of all superpixels in the reference frame over all images. As the boundaries of superpixels follow likely object boundaries, they will reoccur consistently along the true object boundaries across multiple images. The initial shape model (fig. 4) already contains a rough outline of the unknown object class.

At each iteration, we update the parameters of the shape model using the current segmentation of all images according to the maximum likelihood criterion: for each of the 5 orientations in the  $32 \times 32$  grid, we reestimate the empirical probability for a label-change at this position and with this orientation.

While the shape model only knows about the boundaries of an object but not on which side is foreground or background, jointly with the location model (and with the between-image smoothness) it will encourage similar shapes in similar spatial arrangements to be segmented in all the images.

## 4.3 Appearance Model

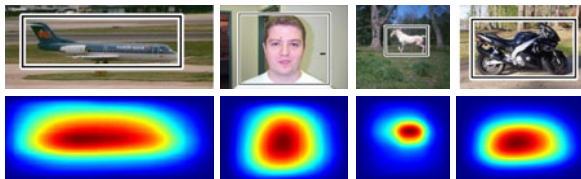
The parameters of the appearance models  $\mathcal{Y}_f$  are initialized using the color/SURF observations from all images using an initial segmentation. This initial segmentation is obtained from a generic prior of object location trained on an external set of images with objects of other classes and their ground-truth segmentations (fig. 3b). From this object location prior, we select the top 75% pixels as foreground; the remaining 25% as background. We observe that this location prior is essentially a Gaussian in the middle of the reference frame.

In each iteration the  $\mathcal{Y}_f$  are updated according to the current segmentations like the location and shape models.

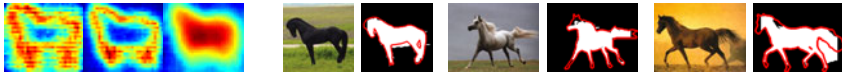
If we are using automatically determined reference frames, the observations for the background are collected from both pixels outside the reference frame and pixels inside the reference frame but labelled as background.

## 5 Finding the Reference Frame

To find the reference frame, we use the objectness measure of [18] which quantifies how likely it is for an image window to contain an object of *any* class. Objectness is trained to distinguish windows containing an object with a well-defined boundary and center, such as cows and telephones, from amorphous



**Fig. 5. Finding the reference frame.** Images with automatically determined reference frames (top) and the objectness maps (bottom).



**Fig. 6. Results on the Weizmann horses.** From left to right: initial shape model, shape model after convergence, location model after convergence, three example images with their segmentations. The ground-truth segmentation is shown in red.

background windows, such as grass and road. Objectness combines several image cues measuring distinctive characteristics of objects, such as appearing different from their surroundings, having a closed boundary, and sometimes being unique within the image.

We sample 1000 windows likely to contain an object from this measure, project the object location prior (sec. 4.3) into these windows and accumulate into an objectness map  $\mathcal{M}$  (fig. 5, (bottom)).  $\mathcal{M}$  will have peaks on the objects in the image. We apply a fixed threshold to  $\mathcal{M}$  and then determine a tight bounding-box around the selected pixels, which we use as the reference frame in our method (fig. 5 (top)).

In the experiments we demonstrate that this method improves the results of unsupervised segmentation compared to using the full images (sec. 6).

## 6 Experiments

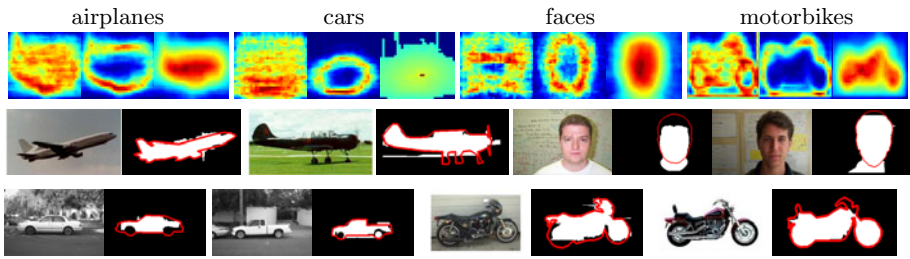
We evaluate the segmentation accuracy of our method as the percentage of pixels classified correctly as either foreground (1) or background (0).

### 6.1 Datasets

We evaluate our unsupervised segmentation method on three datasets of varying difficulty and compare the results to a single-image GrabCut and to other state-of-the-art methods. In no experiment training images with segmentations of the unknown class are used.

**Setting Parameters.** The general parameters to be determined for our model are the weights  $w$  and the generic object location prior. These are determined on external data, i.e. images showing objects of different classes than the one under consideration for unsupervised segmentation (see below for the exact setups). We find weights  $w$  by maximizing segmentation performance on this external data. The weights are optimized using a grid-search on the weight space with the option to switch off individual terms.

**Weizmann Horses [8].** We use the experimental setup of [2]: given 327 images with a horse, segment the horse in each image without using any training images with already segmented horses. Note that other approaches using the Weizmann horses typically use ground-truth segmentations in some of the images for training, e.g. [7]. The weights and the generic object location prior for these experiments are determined from the Caltech4 dataset (as discussed above).



**Fig. 7. Results on Caltech4.** Top row: the initial shape model as well as the shape model and the location model after convergence. Below: for each class, two examples and their segmentations. The ground-truth segmentation is shown in red.

**Caltech4** [27]. We use the experimental setup of [9]: for the classes airplanes, car (sideviews), faces, and motorbikes, we use the test images of [27] and segment the objects using no training data<sup>1</sup>. Weights and generic object location prior are set from the Weizmann Horses dataset.

**Caltech101** [28]. We use an experimental setup similar to [2]: for 28 classes, we randomly select 30 images each and determine the segmentations of the objects. Note that [2] additionally uses 30 training images for each class and solves a joint segmentation and classification task (not done here). Weights and generic object location prior are set by leaving-one-out (setting parameters on 27 classes, and testing on the remaining 1; do this 28 times).

Note that most papers on unsupervised segmentation [2, 8, 10, 13] use variants of these datasets. However, a few object discovery methods, e.g. [11, 29], evaluate on the more difficult datasets.

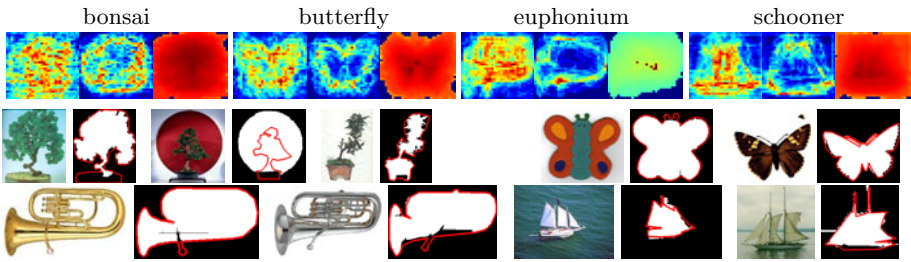
## 6.2 Baselines and the State of the Art

We compare our method to GrabCut [1]. To initialize GrabCut, we train a foreground color model from the central 25% of the area of the image and a background model from the rest. Using these models, GrabCut is iterated until convergence for each image individually. On Weizmann and Caltech4, we evaluate GrabCut in two different setups: (1) using the full image (Tab. 1, line (c)), (2) using the reference frame found by the method in sec. 5 instead of the full image (Tab. 1, line (d)). On Caltech101, we always use the full image as the objects are rather centered. Notice how the automatic reference frame improves the results of GrabCut from line (c) to (d) and how GrabCut is a strong competitor for previous methods [2, 9] that were designed for unsupervised segmentation.

For the datasets for which results are available, we compare our approach to Spatial Topic Models [2] (Tab. 1, line (a)) and to the approach of Arora et al. [9] (Tab. 1, line (b)).

<sup>1</sup> Ground-truth segmentations of the images for quantitative evaluation are taken from the Caltech101 dataset

[http://www.vision.caltech.edu/Image\\_Datasets/Caltech101/](http://www.vision.caltech.edu/Image_Datasets/Caltech101/)



**Fig. 8. Results on Caltech101.** Top row: the initial shape model as well as the shape model and the location model after convergence for four example classes. Below: for each of these classes, some examples with their segmentation. The ground-truth segmentation is shown in red.

We also report the upper bound on the performance that ClassCut can obtain using superpixels [25] (Tab. 1, line (g)). This upper bound corresponds to labeling each superpixel by the majority ground-truth label of its pixels. As the upper bound is always higher than any method we consider, the superpixels are not a limiting factor for the segmentation accuracy of ClassCut.

### 6.3 ClassCut

We evaluate the ability of ClassCut to segment objects of an unknown class in a set of images. Qualitatively, the weights determined show that all terms in our model aid the segmentation process, as none was assigned weight 0. Furthermore, the weights are similar across all setups.

Interestingly, on the Weizmann Horses the GrabCut baseline considering only one image at a time (Tab. 1, line (c)) outperforms the (more complex) spatial topic model [2] (line (a)). When GrabCut is applied within the automatically determined reference frames (line (d)), the result is further improved. ClassCut (line (f)) improves the result a little further. Note also, how ClassCut improves its accuracy over iterations (line (e) to (f)), showing that it is properly learning about the class.

On Caltech4, we compare to [9] (line (b)). Again, the GrabCut baseline is improved when using the automatically determined reference frame rather than the entire image (line (c)/line (d)). This holds even for the classes where the automatically determined reference frames contain a considerable amount of background (cars, faces). ClassCut (line (f)) considerably improves over GrabCut (line (d)) for all classes and on average performs about as well as [9] (ClassCut: 90.6 / [9]: 90.9). Again, ClassCut improves over iterations (from (e) to (f)).

As described above, on Caltech101 we use the full images as reference frames. Using ClassCut we obtain a segmentation accuracy of 83.6%, outperforming both GrabCut (line (c)) and the spatial topic model [2] (line (a)).

Additionally, we evaluate our results using the normalized Chamfer distance to assess how well the segmentation masks align with the shape of the objects. The

**Table 1. Results** are reported as percentage of pixels classified correctly into either foreground or background

Method	Weizmann	Caltech4				Caltech101
	horses	airp.	cars	faces	motorb.	average
(a) Spatial Topic Model [2]	81.8	–	–	–	–	67.0
(b) Arora et al. [9]	–	<b>93.1</b>	<b>95.1</b>	<b>92.4</b>	83.1	–
(c) GrabCut (full image)	83.9	84.5	45.1	83.7	82.4	81.5
(d) GrabCut (reference frames)	85.8	88.7	81.4	89.6	82.3	–
(e) ClassCut (init)	84.7	88.4	90.7	85.3	89.2	83.0
(f) ClassCut (final)	<b>86.2</b>	<b>89.8</b>	<b>93.1</b>	<b>89.0</b>	<b>90.3</b>	<b>83.6</b>
(g) upper bound	92.4	95.5	97.2	93.3	94.7	92.9

Chamfer distance measures the average distance of every point on the segmentation outline to its closest point on the ground-truth outline, normalized by the diagonal of the ground-truth bounding-box. Since neither [2, 9] use any such measure we compare to the GrabCut baseline. For Weizmann/Caltech4/Caltech101 datasets the Chamfer distance averaged over all images is 0.09/0.06/0.13 for ClassCut and 0.20/0.27/0.23 for the corresponding GrabCut baselines. This shows that the segmentations obtained using ClassCut are better aligned to the ground-truth segmentation than those from GrabCut.

## 7 Conclusion

We presented a novel approach to unsupervised class segmentation. Our approach alternates between jointly segmenting the objects in all images and updating a class model, which allows to benefit from the insights gained in interactive segmentation and object class detection. Our model comprises inter-image priors and a comprehensive class model accounting for object appearance, shape, and location w.r.t. an automatically determined reference frame. We demonstrate that the reference frame allows to learn a novel type of shape model and aids the segmentation process.

## References

1. Rother, C., Kolmogorov, V., Blake, A.: GrabCut: interactive foreground extraction using iterated graph cuts. SIGGRAPH 23, 309–314 (2004)
2. Cao, L., Li, F.F.: Spatially coherent latent topic model for concurrent segmentation and classification of objects and scene. In: ICCV (2007)
3. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In: ICCV (2001)
4. Lempitsky, V., Kohli, P., Rother, C., Sharp, T.: Image segmentation with a bounding box prior. In: ICCV (2009)
5. Schroff, F., Criminisi, A., Zisserman, A.: Object class segmentation using random forests. In: Proceedings of the British Machine Vision Conference (2008)

6. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling appearance, shape and context. *IJCV* 81, 2–23 (2009)
7. Kumar, M.P., Torr, P.H.S., Zisserman, A.: OBJ CUT. In: *CVPR* (2005)
8. Borenstein, E., Ullman, S.: Learning to segment. In: Pajdla, T., Matas, J(G.) (eds.) *ECCV 2004. LNCS*, vol. 3023, pp. 315–328. Springer, Heidelberg (2004)
9. Arora, H., Loeff, N., Forsyth, D., Ahuja, N.: Unsupervised segmentation of objects using efficient learning. In: *CVPR* (2007)
10. Winn, J., Jojic, N.: LOCUS: learning object classes with unsupervised segmentation. In: *ICCV* (2005)
11. 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)
12. Todorovic, S., Ahuja, N.: Extracting subimages of an unknown category from a set of images. In: *CVPR* (2006)
13. Galleguillos, C., Babenko, B., Rabinovich, A., Belongie, S.: Weakly supervised object localization with stable segmentations. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part I. LNCS*, vol. 5302, pp. 193–207. Springer, Heidelberg (2008)
14. Kolmogorov, V., Rother, C.: Minimizing nonsubmodular functions with graph cuts – a review. *PAMI* 29, 1274–1279 (2007)
15. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. *PAMI* 28, 1568–1583 (2006)
16. Dalal, N., Triggs, B.: Histogram of Oriented Gradients for Human Detection. In: *CVPR* (2005)
17. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. *PAMI* (2009) (in press)
18. Alexe, B., Deselaers, T., Ferrari, V.: What is an object?. In: *CVPR* (2010)
19. Fulkerson, B., Vedaldi, A., Soatto, S.: Localizing objects with smart dictionaries. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part I. LNCS*, vol. 5302, pp. 179–192. Springer, Heidelberg (2008)
20. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: *CVPR*, vol. 2, pp. 264–271 (2003)
21. Rother, C., Kolmogorov, V., Minka, T., Blake, A.: Cosegmentation of image pairs by histogram matching - incorporating a global constraint into MRFs. In: *CVPR* (2006)
22. Rother, C., Kolmogorov, V., Lempitsky, V., Szmummer, M.: Optimizing binary MRFs via extended roof duality. In: *CVPR* (2007)
23. Csurka, G., Bray, C., Dance, C., Fan, L.: Visual categorization with bags of keypoints. In: *ECCV Workshop on Stat. Learn. in: Comp. Vis.* (2004)
24. Bay, H., Ess, A., Tuytelaars, T., van Gool, L.: SURF: Speeded up robust features. *CVIU* 110, 346–359 (2008)
25. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. *IJCV* 59, 167–181 (2004)
26. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color and texture cues. *PAMI* 26, 530–549 (2003)
27. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: *CVPR* (2003)
28. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In: *IEEE CVPR Workshop of Generative Model Based Vision* (2004)
29. Lee, Y.J., Grauman, K.: Collect-cut: Segmentation with top-down cues discovered in multi-object images. In: *CVPR* (2010)

# A Dynamic Programming Approach to Reconstructing Building Interiors

Alex Flint, Christopher Mei, David Murray, and Ian Reid

Dept. Engineering Science,  
University of Oxford,  
Parks Road, Oxford, UK  
{alexfl, cmei, dwm, ian}@robots.ox.ac.uk

**Abstract.** A number of recent papers have investigated reconstruction under Manhattan world assumption, in which surfaces in the world are assumed to be aligned with one of three dominant directions [1,2,3,4]. In this paper we present a dynamic programming solution to the reconstruction problem for “indoor” Manhattan worlds (a sub-class of Manhattan worlds). Our algorithm deterministically finds the global optimum and exhibits computational complexity linear in both model complexity and image size. This is an important improvement over previous methods that were either approximate [3] or exponential in model complexity [4]. We present results for a new dataset containing several hundred manually annotated images, which are released in conjunction with this paper.

## 1 Introduction

In this paper we investigate the problem of reconstructing simple geometric models from single images of indoor scenes. These scene models can be used to distinguish objects from background in recognition tasks, or provide strong global contextual cues about the observed scene (*e.g.* office spaces, bedrooms, corridors, *etc.*). Point clouds provided by structure-from-motion algorithms are often sparse and do not provide such strong indicators. Compared to a full dense reconstruction, the approach is computationally more efficient and is less sensitive to large texture-less regions typically encountered in indoor environments.

The past few years have seen considerable interest in the Manhattan world assumption [1,2,4,3,5], in which each surface is assumed to have one of three possible orientations. Making this assumption introduces regularities that can improve the quality of the final reconstruction [3]. Several papers have also investigated *indoor* Manhattan models [4,5,6] (a sub-class of Manhattan models), which consist of vertical walls extending between the floor and ceiling planes. A surprisingly broad set of interesting environments can be modelled exactly or approximately as indoor Manhattan scenes [5]. It is with this class of scenes that this paper is concerned.

The present work describes a novel and highly efficient algorithm to obtain models of indoor Manhattan scenes from single images using dynamic programming. In contrast to point cloud reconstructions, our algorithm assigns semantic

labels such as “floor”, “wall”, or “ceiling”. We show that our method produces superior results when compared to previous approaches. Furthermore, our algorithm exhibits running time linear in both image size and model complexity (number of corners), whereas all previous methods that we are aware of [4,5] are exponential in model complexity.

The remainder of the paper is organised as follows. Section 2 describes previous work in this area and section 3 outlines our approach. In section 4 we pose the indoor Manhattan problem formally, then in section 5 we develop the dynamic programming solution. We present experimental results in section 6, including a comparison with previous methods. Concluding remarks are given in the final section.

## 2 Background

Many researchers have investigated the problem of recovering polyhedral models from line drawings. Huffman [7] detected impossible objects by discriminating concave, convex, and occluding lines. Waltz [8] investigated a more general problem involving incomplete line drawings and spurious measurements. Sugihara [9] proposed an algebraic approach to interpreting line drawings, while the “origami world” of Kanade [10] utilised heuristics to reconstruct composites of shells and sheets.

Hoiem *et al.* [11] and Saxena *et al.* [12] have investigated the single image reconstruction problem from a machine learning perspective. Their approaches assign pixel-wise orientation labels using appearance characteristics of outdoor scenes. Hedau *et al.* [6] extend this to indoor scenes, though their work is limited to rectangular box environments.

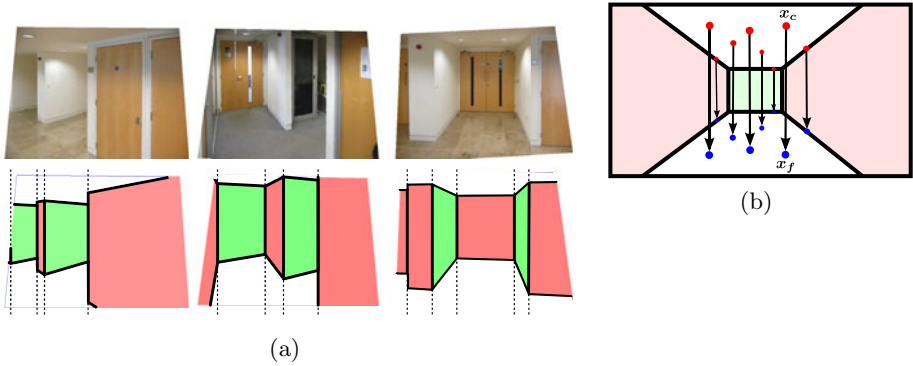
The work most closely related to our own is that of Lee *et al.* [4], which showed that line segments can be combined to generate indoor Manhattan models. In place of their branch-and-bound algorithm, our system uses dynamic programming to efficiently search *all* feasible indoor Manhattan models (rather than just those generated by line segments). As a result we obtain more accurate models, can reconstruct more complex environments, and obtain computation times several orders of magnitude faster than their approach, as will be detailed in Section 6.

Furukawa *et al.* [3] used the Manhattan world assumption for stereo reconstruction. They make use of multiple calibrated views, and they search a different class of models, so their approach is not comparable to ours.

Barinova *et al.* [13] suggested modelling outdoor scenes as a series of vertical planes. Their models bear some similarity to ours but they cannot handle occluding boundaries, and their EM inference algorithm is less efficient than our dynamic programming approach.

Felzenszwalb and Veksler [14] applied dynamic programming to a class of pixel labelling problems. Because they optimise directly in terms of pixel labels their approach is unable to capture the geometric feasibility constraints that our system utilises.





**Fig. 1.** (a) Three input images and the indoor Manhattan models we seek. Notice how each image column intersects exactly one wall. (b) The mapping  $H_{c \rightarrow f}$  transfers points between the ceiling and floor.

### 3 Outline of Proposed Approach

Our goal is to reconstruct an indoor Manhattan model from a single image. Three example images and the models we seek for them are shown in Figure 1a. A perfectly uncluttered environment such as that shown in the third column of Figure 1a could be represented exactly by an indoor Manhattan model, though in general we expect to encounter clutter and our goal in such cases is to recover the boundaries of the environment in spite of this distraction. That is, we aim to completely ignore all objects within the room and reconstruct the bare room boundaries, in contrast to most previous approaches that aim to reconstruct the entire scene. This choice is due to our intention of using the models as input for further reasoning.

The Manhattan world assumption states that world surfaces are oriented in one of three mutually orthogonal directions [1]. The indoor Manhattan assumption further states that the environment consists of a floor plane, a ceiling plane, and a set of walls extending vertically between them [4]. Each wall therefore has one of two possible orientations (ignoring sign), and each corner<sup>1</sup> is either concave, convex, or occluding, as depicted in Figure 3. Indoor Manhattan models are interesting because they can represent many indoor environments approximately or exactly, yet they introduce regularities to the reconstruction problem that makes possible a left-to-right decomposition of the scene, on which the dynamic programming algorithm developed in this paper rests. Our approach to reconstructing indoor Manhattan environments consists of the following five steps:

1. Identify three dominant surface orientations. (Section 3.1)
2. Identify the floor and ceiling planes. (Section 3.2)

<sup>1</sup> We use “corner” throughout this paper to refer to the intersection of two walls, which appears as a line segment in the image.

3. Rectify vertical lines. (Section 3.3)
4. Obtain weak orientation estimates. (Section 3.4)
5. Estimate the final model. (Sections 4 and 5)

### 3.1 Identifying Dominant Directions

We identify three dominant directions by estimating mutually orthogonal vanishing points in the image. Our approach is similar to Koseckà and Zhang [2], in which  $k$ -means clustering provides an initial estimate that is refined using EM. We assume that the vertical direction in the world corresponds to the vanishing point with largest absolute  $y$ -coordinate, which we label  $\mathbf{v}_v$ . The other two vanishing points are denoted  $\mathbf{v}_l$  and  $\mathbf{v}_r$ .

If the camera intrinsics are unknown then we construct the camera matrix  $K$  from the detected vanishing points by assuming that the camera centre is at the image centre and choosing a focal length and aspect ratio such that the calibrated vanishing points are mutually orthogonal.

### 3.2 Identifying the Floor and Ceiling Planes

An indoor Manhattan scene has exactly one floor and one ceiling plane, both with normal direction  $\mathbf{v}_v$ . It will be useful in the following sections to have available the mapping  $H_{c \rightarrow f}$  between the image locations of ceiling points and the image locations of the floor points that are vertically below them (see Figure 1b).  $H_{c \rightarrow f}$  is a planar homology with axis  $\mathbf{h} = \mathbf{v}_l \times \mathbf{v}_r$  and vertex  $\mathbf{v}_v$  [15] and can be recovered given the image location of any pair of corresponding floor/ceiling points  $(\mathbf{x}_f, \mathbf{x}_c)$  as

$$H_{c \rightarrow f} = I + \mu \frac{\mathbf{v}_v \mathbf{h}^T}{\mathbf{v}_v \cdot \mathbf{h}}, \quad (1)$$

where  $\mu = \langle \mathbf{v}_v, \mathbf{x}_c, \mathbf{x}_f, \mathbf{x}_c \times \mathbf{x}_f \times \mathbf{h} \rangle$  is the characteristic cross ratio of  $H_{c \rightarrow f}$ .

Although we do not have *a priori* any such pair  $(\mathbf{x}_f, \mathbf{x}_c)$ , we can recover  $H_{c \rightarrow f}$  using the following RANSAC algorithm. First, we sample one point  $\hat{\mathbf{x}}_c$  from the region above the horizon in the Canny edge map, then we sample a second point  $\hat{\mathbf{x}}_f$  collinear with the first and  $\mathbf{v}_v$  from the region below the horizon. We compute the hypothesis map  $\hat{H}_{c \rightarrow f}$  as described above, which we then score by the number of edge pixels that  $\hat{H}_{c \rightarrow f}$  maps onto other edge pixels (according to the Canny edge map). After repeating this for a fixed number of iterations we return the hypothesis with greatest score.

Many images contain either no view of the floor or no view of the ceiling. In such cases  $H_{c \rightarrow f}$  is unimportant since there are no corresponding points in the image. If the best  $H_{c \rightarrow f}$  output from the RANSAC process has a score below a threshold  $k_t$  then we set  $\mu$  to a large value that will transfer all pixels outside the image bounds.  $H_{c \rightarrow f}$  will then have no impact on the estimated model.

### 3.3 Rectifying Vertical Lines

The algorithms presented in the remainder of this paper will be simplified if vertical lines in the world appear vertical in the image. We therefore warp images according to the homography

$$H = \begin{pmatrix} \mathbf{v}_v \times \mathbf{e}_3 \\ \mathbf{v}_v \\ \mathbf{v}_v \times \mathbf{e}_3 \times \mathbf{v}_v \end{pmatrix}, \quad \mathbf{e}_3 = [0, 0, 1]^T. \quad (2)$$

### 3.4 Obtaining Weak Orientation Estimates

Our algorithm requires a pixel-wise surface orientation estimate to bootstrap the search. Obtaining such estimates has been explored by several authors [4,11,12]. We adopt the simple and efficient line-sweep approach of Lee *et al.* [4], which produces a partial labelling of the image in terms of the three Manhattan surface orientation labels (corresponding to the three Manhattan orientations in the image). We denote this orientation map  $o : \mathbb{Z}^2 \rightarrow \{l, r, v, \emptyset\}$  where  $\emptyset$  represents the case in which no label is assigned and  $l, r$ , and  $v$  correspond to the three vanishing points  $\{\mathbf{v}_l, \mathbf{v}_r, \mathbf{v}_v\}$ .

Note that our algorithm is not dependent on the manner in which  $o$  is obtained; any method capable of estimating surface orientations from a single image, including the work of Hoiem [11] or Saxena [12], could be used instead.

We generate three binary images  $B_a$ ,  $a \in \{l, r, v\}$  such that  $B_a(\mathbf{x}) = 1$  if and only if  $o(\mathbf{x}) = a$ . We then compute the integral image for each  $B_a$ , which allows us to count the number of pixels of a given orientation within any rectangular sub-image in  $O(1)$  time. This representation expedites evaluation of the cost function described later.

## 4 Formulation of Reconstruction Problem

Consider the indoor Manhattan scenes shown in Figure 1a. Despite the complexity of the original images, the basic structure of the scene as depicted in the bottom row is simple. In each case there is exactly one wall between any two adjacent corners<sup>1</sup>, so any vertical line intersects at most one wall. This turns out to be a general property of indoor Manhattan environments that arises because the camera must be between the floor and ceiling planes. Any indoor Manhattan scene can therefore be represented as a series of one or more wall segments in order from left to right.

Given the warp performed in the Section 3.3, corners are guaranteed to appear vertical in the image, so can be specified simply by an image column. Furthermore, given the mapping  $H_{c \rightarrow f}$  from Section 3.2 the image location of either the top or bottom end-point of a corner (*i.e.* the intersection of the wall with the ceiling or floor respectively) is sufficient to specify both and thereby the line segment representing that corner. Without loss of generality we choose to represent corners by their upper end-point. A wall segment can then be specified

by its left and right corners, together with its associated vanishing point (which must be either  $\mathbf{v}_l$  or  $\mathbf{v}_r$ ), as illustrated in Figure 2a.

This leads to a simple and general parametrisation under which we represent an indoor Manhattan model  $\mathcal{M}$  as an alternating sequence of corners and walls  $(c_1, W_1, c_2, W_2, \dots, W_{n-1}, c_n)$ ,  $c_i < c_{i+1}$ . Each corner  $c_i$  is the column index at which two walls meet, and each wall  $W_i = (r_i, a_i)$  comprises an orientation  $a_i \in \{l, r\}$ , which determines whether its vanishing point is  $\mathbf{v}_l$  or  $\mathbf{v}_r$ , and a row index  $r_i$ , at which its upper edge meets the corner to its left (see Figure 2b). Hence the upper-left corner of the  $i^{\text{th}}$  wall is  $(c_i, r_i)$ , which, together with its vanishing point  $\mathbf{v}_{a_i}$ , fully specifies its location in the image. Clockwise from top-left the vertices of the  $i^{\text{th}}$  wall are

$$\mathbf{p}_i = [c_i, r_i, 1]^T, \quad \mathbf{q}_i = \mathbf{p}_i \times \mathbf{v}_{a_i} \times [1, 0, -c_{i+1}]^T, \quad \mathbf{r}_i = H_{c \rightarrow f} \mathbf{q}_i, \quad \mathbf{s}_i = H_{c \rightarrow f} \mathbf{p}_i. \quad (3)$$

A model  $\mathcal{M}$  generates for each pixel  $\mathbf{x}$  a predicted surface orientation  $g_{\mathcal{M}}(\mathbf{x}) \in \{l, r, v\}$  corresponding to one of the three vanishing points  $\{\mathbf{v}_l, \mathbf{v}_r, \mathbf{v}_v\}$ . We compute  $g_{\mathcal{M}}$  by filling quads corresponding to each wall segment, then filling the remaining area with the floor/ceiling label  $v$ .

Not all models  $\mathcal{M}$  are physically realisable, but those that are not can be discarded using simple tests on the locations of walls and vanishing points as enumerated by Lee *et al.* [4]. The reader is referred to their paper for details; the key result for our purposes is that a model is feasible if all of its corners are feasible, and the feasibility of a corner is dependent only on the immediately adjoining walls.

### 4.1 Formalisation

We are now ready to formalise the minimisation problem. Given an input image of size  $W \times H$  and an initial orientation estimate  $o$ , the pixel-wise cost  $C_d(\mathbf{x}, a)$  measures the cost of assigning the label  $a \in \{l, r, v\}$  to pixel  $\mathbf{x}$ . We adopt the simple model,

$$C_d(\mathbf{x}, a) = \begin{cases} 0, & \text{if } o(\mathbf{x}) = a \text{ or } o(\mathbf{x}) = \emptyset \\ 1, & \text{otherwise} \end{cases}. \quad (4)$$

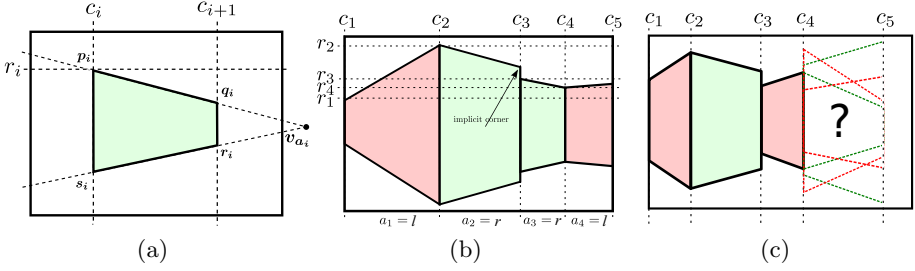
The cost for a model  $\mathcal{M}$  consisting of  $n$  corners is then the sum over pixel-wise costs,

$$C(\mathcal{M}) = n\lambda + \sum_{\mathbf{x} \in \mathcal{I}} C_d(\mathbf{x}, \mathcal{M}(\mathbf{x})) \quad (5)$$

where  $\lambda$  is a constant and  $n\lambda$  is a regularisation term penalising over-complex models. We seek the model with least-cost

$$\mathcal{M}^* = \underset{\mathcal{M}}{\operatorname{argmin}} C(\mathcal{M}). \quad (6)$$

where implicit in (5) is the restriction to labellings representing indoor Manhattan models, since only such labellings can be represented as models  $\mathcal{M}$ . Figure 1a shows optimal models  $\mathcal{M}^*$  for three input images.



**Fig. 2.** (a) The row/column indices  $c_i, c_{i+1}, r_i$ , together with the vanishing point index  $a_i \in \{l, r\}$  are sufficient via the homology  $H_{c \rightarrow f}$  to determine the four vertices defining a wall. (b) An illustration of the model  $\mathcal{M} = \{c_1, (r_1, a_1), \dots, c_4, (r_4, a_4), c_5\}$ . (c) A partial model covering columns to  $c_4$  with several feasible (green dashed) and infeasible (red dashed) wall segments.

## 5 Proposed Algorithm

In this section we present a dynamic programming solution to the minimisation problem posed in the previous section. We develop the algorithm conceptually first, then formalise it later.

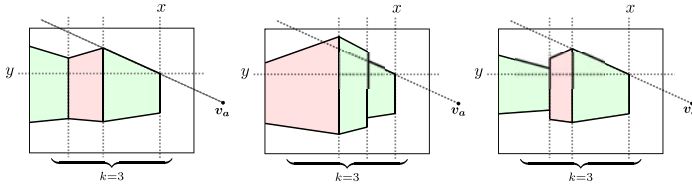
We have already seen that every indoor Manhattan scene can be represented as a left-to-right sequence of wall segments, and every image column intersects exactly one wall segment. As a result, the placement of each wall is “conditionally independent” of the other walls given its left and right neighbours. For example, Figure 2c shows a partial model as well as several wall segments that could be appended to it. Some of the candidates are feasible (green dashes) and some are not (red dashes); however, note that once the wall segment from  $c_3$  to  $c_4$  is fixed, the feasibility of wall segments following  $c_4$  is independent of choices made for wall segments prior to  $c_3$ .

This leads to a decomposition of the problem into a series of sub-problems of the form “find the minimum-cost *partial* model that terminates<sup>2</sup> at  $\mathbf{x} = (c, r)$ ”. To solve this we enumerate over all possible walls  $W$  that have top-right corner at  $\mathbf{x}$ , then for each we recursively solve the sub-problems for the partial model terminating at each  $\mathbf{x}'$ , where  $\mathbf{x}'$  ranges along the left edge of  $W$ . This recursive process eventually reaches the left boundary of the image since  $\mathbf{x}'$  is always strictly to the left of  $\mathbf{x}$ , at which point the recursion terminates. As is standard in dynamic programming approaches, the solution to each sub-problem is cached to avoid redundant computation. To solve the complete minimisation (6) we simply solve the sub-problems corresponding to each point on the right boundary of the image.

We now formalise the dynamic programming algorithm. Let  $f_{in}(x, y, a, k)$  be the cost of a model  $\mathcal{M}^+ = \{c_1, W_1, \dots, W_{k-1}, c_k\}$  such that

1.  $c_k = x$  (i.e. the model terminates at column  $x$ ),
2.  $W_{k-1} = (y, a)$  (i.e. the model terminates at row  $y$  with orientation  $a$ ),

<sup>2</sup> A model “terminates” at the top-right corner of its right-most wall.



**Fig. 3.** Three models satisfying constraints 1–3 for the sub–problem  $f_{in}(x, y, a, k)$ . Only one will satisfy the least–cost constraint.

3.  $\mathcal{M}^+$  is feasible, and
4.  $\mathcal{M}^+$  has minimal cost among all such models.

We show in the additional material that if a model

$$\mathcal{M} = \{c_1, W_1, \dots, W_{k-1}, c_k\} \tag{7}$$

is a solution to the sub–problem  $f_{in}(c_k, r_k, a_k, k)$ , then the truncated model

$$\mathcal{M}' = \{c_1, W_1, \dots, W_{k-2}, c_{k-1}\} \tag{8}$$

is a solution to the sub–problem  $f_{in}(c_{k-1}, r_{k-1}, a_{k-1}, k - 1)$ . In light of this we introduce the following recurrence relation:

$$f_{in}(x, y, a, k) = \min_{x' < x, y', a'} \left( f_{in}(x', y', a', k - 1) + C_w \right), \tag{9}$$

where  $C_w$  is the cost of the wall  $W = (y', a')$ , computed by summing  $C_d$  over columns  $x'$  to  $x$ . The minimisation (9) is performed subject to feasibility constraints, so for each  $x' < x$ , only a subset of  $y$ –coordinates are considered. Since a model must have zero or more corners and a model with right–most corner at  $x = 0$  does not span any part of the image, we have the boundary conditions

$$f_{in}(x, y, a, k) = \begin{cases} 0, & \text{if } x = 0 \\ \infty, & \text{if } x \neq 0 \text{ and } k < 0 \end{cases} . \tag{10}$$

Finally, the cost of the optimal model (6) is

$$C(\mathcal{M}^*) = \min_{\substack{1 \leq y \leq H \\ k \leq K \\ a \in \{l, r\}}} \left( f_{in}(W, y, a, k) + \lambda k \right) . \tag{11}$$

where  $K$  is a parameter specifying the maximum model complexity and  $\lambda$  is the per–wall penalty.

We compute  $C(\mathcal{M}^*)$  by recursively evaluating  $f_{in}$  according to (9) until we reach one of the boundary conditions (10). In line with standard dynamic programming theory we cache each evaluation to avoid redundant computation. For each cache entry we also store  $x', y', a'$  corresponding to least–cost wall identified

when evaluating (9), which allows the desired model  $\mathcal{M}^*$  to be reconstructed by back-tracking once all evaluations are complete.

**Complexity.** Due to the caching scheme, (9) is evaluated at most once for each unique set of parameters. There are  $2WHK$  possible parameters and the complexity of each evaluation is  $O(W^2H)$ , since the minimisation in (9) is over  $O(WH)$  terms and computing each marginal cost  $C_w$  requires  $O(W)$  additions<sup>3</sup>. The overall complexity of the basic algorithm is therefore  $O(W^3H^2K) = O(L^5K)$  where  $L = \max(W, H)$ .

### 5.1 Auxiliary Sub-problems

The basic algorithm described thus far involves minimising over all pixels to the left of  $\mathbf{x}$  for each pixel  $\mathbf{x}$ , (i.e. the joint minimisation over  $x'$  and  $y'$  in (9)). In the previous section we enforced feasibility by explicitly testing each  $(x', y')$  and omitting any that would lead to an infeasible model from the minimisation (9). In this section we show that by introducing auxiliary sub-problems that build feasibility into the core of the algorithm we can significantly reduce computational complexity.

We introduce three new sub-problems  $f_{up}$ ,  $f_{down}$ , and  $f_{out}$ . Each is identical to  $f_{in}$  except that constraint 2 is modified as follows:

- $f_{out}$     appending  $W = (y, a)$  to  $M^+$  would produce a feasible model.
- $f_{up}$      $r_{k-1} \leq y$  (i.e. the right-most wall terminates above row  $y$ )
- $f_{down}$     $r_{k-1} \geq y$  (i.e. the right-most wall terminates below row  $y$ )

Consider first the sub-problem  $f_{out}(x, y, a, k)$ , and suppose that the right-most wall in its solution is  $W'$ . Now  $W'$  terminates either above row  $y$ , below row  $y$ , or exactly at row  $y$ , which correspond respectively to the sub-problems  $f_{up}$ ,  $f_{down}$ , and  $f_{in}$ . We also have two choices of orientation, making six possibilities in total, from which we select the one with least cost,

$$f_{out}(x, y, a, k) = \min_{a' \in \{l, r\}} \min \begin{cases} f_{up}(x, y - 1, a', k) \\ f_{in}(x, y, a', k) \\ f_{down}(x, y + 1, a', k) \end{cases}, \quad (12)$$

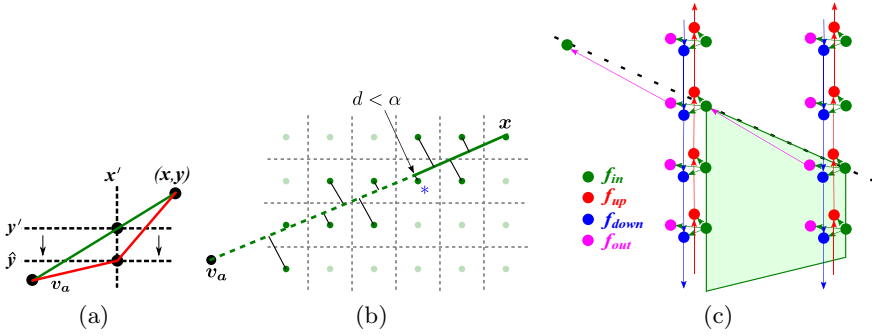
where either or both of the  $f_{up}$  and  $f_{down}$  terms are omitted if such a wall would be infeasible.

Similarly, suppose that the least-cost model that terminates at  $(x, y)$  (i.e. the solution to  $f_{in}(x, y, a, k)$ ) has right-most wall  $W'$ . Now  $W'$  must have its left edge at some column  $x' < x$ , and the portion of the model to the left of  $x'$  must be feasible when  $W'$  is appended. Hence we have

$$f_{in}(x, y, a, k) = \min_{x' < x} \left( f_{out}(x', y', a, k - 1) + C_W \right), \quad (13)$$

where  $y'$  is the  $y$ -coordinate at which the line through  $\mathbf{v}_a$  and  $(x, y)$  meets column  $x'$  and  $C_w$  is the cost of the wall  $W' = (y', a')$  exactly as in (9). Note

<sup>3</sup> Here we use the integral images  $B_i$ .



**Fig. 4.** (a) The bend introduced by rounding  $y'$  to  $\hat{y} = \lfloor y' + 0.5 \rfloor$ . (b) A line from  $x$  to  $v_a$  with the distances  $d$  to nearby pixel centres (green dots). The starred pixel is the first that satisfies  $d < \epsilon$ . (c) A graph in which each node represents a sub-problem and each edge is a dependence relation. Two columns are expanded; other column are omitted for brevity. The green quad is a wall corresponding to a particular pair of nodes in the graph.

that (13) consists of  $O(L)$  terms whereas in the previous section (9) consisted of  $O(L^2)$  terms.

Finally we may decompose the  $f_{up}$  and  $f_{down}$  sub-problems each into two cases,

$$f_{up}(x, y, a, k) = \begin{cases} \min(f_{in}(\cdot), f_{up}(x, y - 1, a, k)), & \text{if } y \geq 1 \\ \infty, & \text{otherwise} \end{cases} \quad (14)$$

$$f_{down}(x, y, a, k) = \begin{cases} \min(f_{in}(\cdot), f_{down}(x, y + 1, a, k)), & \text{if } y \leq H \\ \infty, & \text{otherwise} \end{cases} \quad (15)$$

The dependencies between the sub-problems are illustrated as an evaluation graph in Figure 4c.

### 5.2 From $(L^3K)$ to $O(L^2K)$

Evaluating (13) remains an  $O(W)$  operation due to the minimisation over  $x'$ . In this section we reduce this to  $O(1)$ .

Consider the sub-problem  $f_{in}(x, y, a, k)$  as formulated in the previous section. Evaluating  $f_{in}$  is like walking along each column  $x - 1, x - 2, \dots, 1$  and considering two possibilities at each step: insert a corner or continue walking. The former corresponds to evaluating  $f_{out}(x', y', a, k - 1) + C_w$ ; that is, we insert a wall between  $x'$  and  $x$  with cost  $C_w$ , then find the optimal model that occupies the remaining space to the left of  $x'$ . The latter corresponds to evaluating  $f_{in}(x', y', a, k) + C_w$ ; that is, we find the best model that terminates at  $(x', y')$  with orientation  $a$  and extend its right-most wall to  $(x, y)$ . But  $y'$  is computed by intersecting the line from  $v_a$  to  $(x, y)$  with image column  $x'$ , so in general  $y'$  is not an integer. While



it is sufficient to round  $y'$  to the nearest integer  $\hat{y} = \lfloor y' + 0.5 \rfloor$  when evaluating  $f_{out}$ , doing the same for  $f_{in}$  would produce a bend in the wall as shown in Figure 4a. In (13) we avoided this by evaluating  $f_{out}$  for all  $x' < x$ , but this is unnecessarily wasteful. We now introduce a threshold  $\epsilon$  and allow  $\hat{y}$  to replace  $y'$  whenever

$$|y' - \hat{y}| < \epsilon . \quad (16)$$

When we encounter an image column satisfying (16) we evaluate  $f_{in}$  as follows. We consider adding a corner at  $x'$  by evaluating  $f_{out}(x', y', a, k - 1) + C_w$  as in the previous section, then we consider the case that the wall continues past column  $x'$  by evaluating  $f_{in}(x', \hat{y}, a, k) + C_w$ , and we return the minimum of the two values. At this point we need not consider any further columns to the left of  $x'$  since any such consideration are already captured in the evaluation of  $f_{in}(x', \hat{y}, a, k)$ . Hence rather than evaluating all  $x' < x$  we need only walk as far as the first  $x'$  that satisfies (16), as shown in Figure 4b. The recurrence relation for  $f_{in}$  now becomes

$$f_{in}(x, y, a, k) = \min \left\{ \begin{array}{l} \min_{x_p \leq x' < x} (f_{out}(x', y', a, k - 1) + C_w) \\ f_{in}(x_p, y_p), a, k - 1) + C_w \end{array} \right. . \quad (17)$$

where  $x_p < x$  is the closest column to  $x$  satisfying (16) and  $y_p$  is the row at that column meets the line from  $(x, y)$  to  $\mathbf{v}_a$ . Empirically we have found that even for  $\epsilon = 0.01$  pixels, we always encounter some  $x'$  satisfying (16) within 20 steps from any start point.

**Complexity.** Evaluating each sub-problem is now an  $O(1)$  operation, so the overall complexity of the algorithm is given by the total number of unique sub-problems, which is

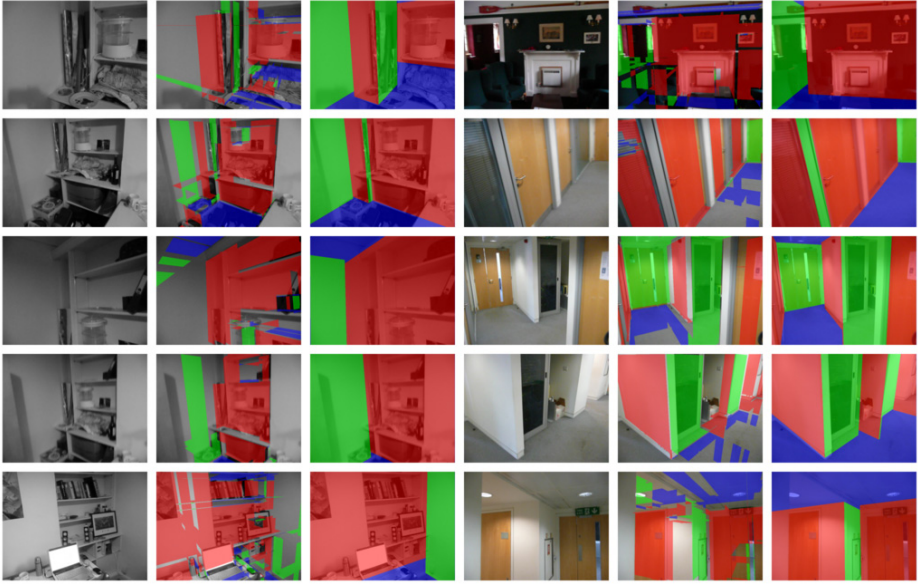
$$O(KL^2) . \quad (18)$$

## 6 Results

We tested our system on a dataset of 634 manually annotated images of indoor scenes. To expedite annotation we collected video sequences and used structure-from-motion software to recover camera poses, allowing us to project a manually specified floor plan into each view.

In each experiment we computed the fraction of pixels for which the orientation predicted by the output model  $\mathcal{M}$  agreed with the ground truth orientation. Unless otherwise specified, the parameter settings for the experiments below are  $\epsilon = 0.01$ ,  $K = 7$ ,  $m = 4$ ,  $\lambda = 100$ . Image sizes were  $640 \times 480$  pixels. We found our algorithm to be robust to all of these parameter values, as the following experiments show.

We compared our results with the branch-and-bound approach of Lee *et al.* [4]. In 138 of the images (21.7% of the dataset), their method was unable to estimate a building model as there was no appropriate pair of line segments with which to initialise their approach. In a pixel-wise evaluation their approach was



**Fig. 5.** Models estimated by our algorithm. Each panel contains three images: the original image, the initial orientation estimate, and the final model output by our system. Best viewed in colour.

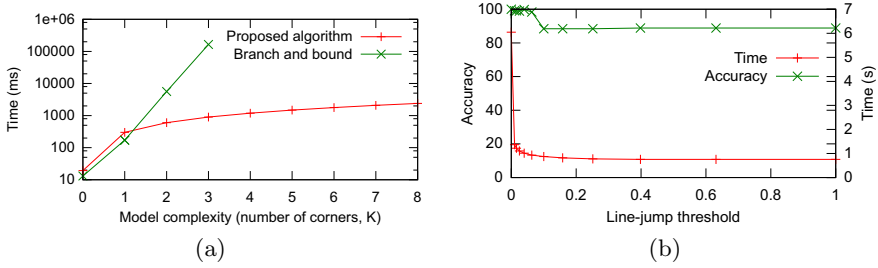


**Fig. 6.** Failure cases of our system. Best viewed in colour.

able to correctly label 54.3% of pixels, while our approach obtained an accuracy of 79.7%. Omitting the images for which their approach was unable to estimate a building structure, their approach obtained 68.1% accuracy. We believe that the difficulty of our dataset (many occluding objects, many images without a view of both floor and ceiling) accounts for the significantly lower performance in comparison to that quoted in [4]. Side-by-side comparisons with their approach are included in additional material.

### 6.1 Failure Cases

Figure 6 shows four representative failure cases of our approach. In the top-left panel the occlusion relationship between two walls is incorrectly estimated,



**Fig. 7.** (a) Efficiency comparison with the branch-and-bound algorithm of Lee *et al.* [4] (our implementation). Their approach scales exponentially with model complexity whereas ours scales only linearly. (b) The line-jump parameter  $\epsilon$  trades off accuracy (diamonds) for computation time (crosses). Accuracy is computed relative to the baseline  $\epsilon = 0$ . Small values of  $\epsilon$  achieve significant speedup with no perceivable degradation in accuracy. Based on these results we set  $\epsilon = 0.01$  in our remaining experiments.

so the more distant wall is thought to be occluding the closer wall. This is because the floor patch in the bottom centre of the image is missed in the initial orientation estimate. In the top-right panel, too few line segments are detected and the initial orientation estimate is very poor. The bottom-left panel shows an example of a chair that is wrongly identified as part of a wall. The chair is aligned with the wall behind it and this highlights the limitation of using only line segments to estimate an initial orientation estimate. The bottom-right panel shows how a deviation from the indoor Manhattan assumption causes an incorrect model to be estimated. The exit sign represents a vertical surface that does not extend from the ceiling to the floor, which our approach is currently unable to handle.

## 7 Discussion

We have shown that semantically meaningful models of indoor scenes can be recovered efficiently for a range of Manhattan environments using dynamic programming. Our approach is able to model complex scenes, which would be intractable for previous methods that involved combinatorial searches in the space of models. This work represents an important increment on the state-of-the-art both in terms of accuracy and efficiency.

An alternative approach might be to apply graph cuts to this problem. However, Kolmogorov and Zabih [16] showed that only regular functions (a subset of sub-modular functions) can be minimised via graph cuts, and the cost (6) is not regular because implicit in the minimisation is the hard constraint that labellings must form an indoor Manhattan model, which induces complicated dependencies between the pixels in each column. Even if an appropriate relaxation of this constraint yielded a regular cost function, applying graph cuts would entail using a technique such as  $\alpha$ -expansion [16], which is both approximate and

non-deterministic. In contrast, our approach is exact, deterministic, and highly efficient.

Future work will investigate richer cues for obtaining the initial orientation estimates as well as a probabilistic formulation of the cost function (4).

## References

1. Coughlan, J., Yuille, A.: Manhattan world: compass direction from a single image by bayesian inference. In: CVPR, vol. 2, pp. 941–947 (1999)
2. Kösecká, J., Zhang, W.: Video compass. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part IV. LNCS, vol. 2353, pp. 476–490. Springer, Heidelberg (2002)
3. Furukawa, Y., Curless, B., Seitz, S., Szeliski, R.: Manhattan-world stereo. In: CVPR, pp. 1422–1429 (2009)
4. Lee, D.C., Hebert, M., Kanade, T.: Geometric reasoning for single image structure recovery. In: CVPR (2009)
5. Flint, A., Mei, C., Reid, I., Murray, D.: Growing semantically meaningful models for visual slam. In: CVPR (2010)
6. Hedau, V., Hoiem, D., Forsyth, D.: Recovering the spatial layout of cluttered rooms. In: ICCV, vol. 2 (2009)
7. Huffman, D.A.: Impossible objects as nonsense sentences. *Machine Intelligence* 6, 295–323 (1971)
8. Waltz, D.L.: Generating semantic descriptions from drawings of scenes with shadows. Technical report, MIT (1972)
9. Sugihara, K.: Mathematical structures of line drawings of polyhedrons. *PAMI* 4, 458–469 (1982)
10. Kanade, T.: A theory of origami world. *Artificial Intelligence* 13, 279–311 (1980)
11. Hoiem, D., Efros, A.A., Hébert, M.: Geometric context from a single image. In: ICCV, pp. 654–661 (2005)
12. Saxena, A., Sun, M., Ng, A.Y.: Make3d: Learning 3d scene structure from a single still image. *PAMI* 31, 824–840 (2009)
13. Barinova, O., Konushin, V., Yakubenko, A., Lee, K., Lim, H., Konushin, A.: Fast automatic single-view 3-d reconstruction of urban scenes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 100–113. Springer, Heidelberg (2008)
14. Felzenszwalb, D., Veksler, O.: Tiered scene labeling with dynamic programming. In: CVPR (2010)
15. Criminisi, A.: *Accurate visual metrology from single and multiple uncalibrated images*. Springer, New York (2001)
16. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? *PAMI* (2002)

# Discriminative Mixture-of-Templates for Viewpoint Classification

Chunhui Gu<sup>1</sup> and Xiaofeng Ren<sup>2</sup>

<sup>1</sup> University of California at Berkeley, Berkeley, CA 94720, USA  
chunhui@eecs.berkeley.edu

<sup>2</sup> Intel Labs Seattle, 1100 NE 45th Street, Seattle, WA 98105, USA  
xiaofeng.ren@intel.com

**Abstract.** Object viewpoint classification aims at predicting an approximate 3D pose of objects in a scene and is receiving increasing attention. State-of-the-art approaches to viewpoint classification use generative models to capture relations between object parts. In this work we propose to use a mixture of holistic templates (e.g. HOG) and discriminative learning for joint viewpoint classification and category detection. Inspired by the work of Felzenszwalb et al 2009, we discriminatively train multiple components simultaneously for each object category. A large number of components are learned in the mixture and they are associated with canonical viewpoints of the object through different levels of supervision, being fully supervised, semi-supervised, or unsupervised. We show that discriminative learning is capable of producing mixture components that directly provide robust viewpoint classification, significantly outperforming the state of the art: we improve the viewpoint accuracy on the Savarese et al 3D Object database from 57% to **74%**, and that on the VOC 2006 car database from 73% to **86%**. In addition, the mixture-of-templates approach to object viewpoint/pose has a natural extension to the continuous case by discriminatively learning a linear appearance model locally at each discrete view. We evaluate continuous viewpoint estimation on a dataset of everyday objects collected using IMUs for groundtruth annotation: our mixture model shows great promise comparing to a number of baselines including discrete nearest neighbor and linear regression.

## 1 Introduction

One fundamental property of visual sensing is that it is a projection process from a 3D world to a 2D image plane; much of the 3D information is lost in the projection. How to model and re-capture the 3D information from 2D views has been at the center of the computer vision research. One classical example is the *aspect graphs* of Koenderink and van Doorn [1], where a 3D object is modeled as a collection of inter-connected 2D views.

A complete understanding of objects in a visual scene comprises not only labeling the identities of objects but also knowing their poses in 3D. Most of the recent vision research has been devoted to the recognition problem, where

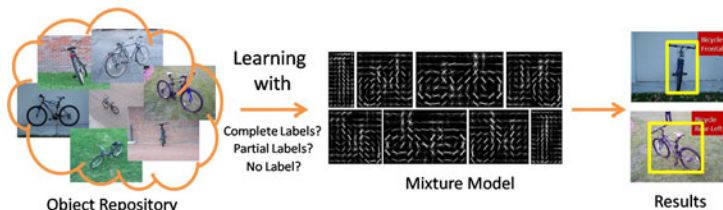
huge progresses have been made: the SIFT matching framework [2] and the HOG models [3,4] are good representatives of how much object recognition capabilities have progressed over the years. The 3D object pose problem have received much less but still considerable attention. The series of work from Savarese and Fei-Fei [5,6,7] are good examples of how people approach the 3D pose problem in modern contexts, where large benchmarks are established and evaluated for discrete viewpoint classification [5,8].

There have been, however, divergent trends between object recognition and pose estimation. Latest progresses in object recognition employ discriminative templates directly trained from image gradients [4]; latest 3D pose models group features into parts and learn generative models of their relationships [6,7].

We believe the two problems should be one and identical, that a good framework of object detection should be able to handle both category and viewpoint classification. In particular, discriminative learning, which has seen great successes in category classification, should readily apply to viewpoint classification.

In this work we present strong empirical proof that it is indeed the case: a discriminatively learned mixture of templates, extending the latent HOG framework of Felzenszwalb et al [4], is capable of representing a large number of viewpoints (as components) and handling both category and viewpoint classification. A mixture-of-HOG model produces superior results for all the three cases of supervised (with groundtruth view labels), semi-supervised (with a subset of view labels) and unsupervised (no view labels) viewpoint learning (see Figure 1). Furthermore, the mixture-of-templates approach has a natural extension to the continuous case: we propose a continuous viewpoint model which linearly approximates local appearance variations at each discrete view. This model is discriminatively trained, just as in the discrete case, and outputs a continuous 3D viewpoint/pose.

We evaluate our approach on a number of 3D object databases, including the 3DObject Database of Savarese [5], the VOC2006 car database [8], and a dataset of our own for benchmarking continuous viewpoint estimation. We



**Fig. 1.** We propose to use a discriminative mixture of templates for object viewpoint classification. We discriminatively learn a large mixture of templates using HOG [3,4] and show that the templates correspond well to the canonical views of an object, which are directly used for viewpoint classification and significantly outperform the state of the art. We show that the mixture model works well when trained with complete viewpoint labels (supervised), a subset of labels (semi-supervised), and no viewpoint labels (unsupervised). We then extend the mixture model for continuous pose prediction, again using a discriminative mixture of templates.

show that we significantly outperform the state-of-the-art results on all these challenging benchmarks: we improve the 8-way viewpoint classification accuracy on the 3DObject database from 57% to 74%, and that on the VOC2006 cars from 73% to 85%. For the continuous case, we show that our discriminative mixture model outperforms a number of baselines, including one using the closest discrete viewpoint and one using linear regression on top of the viewpoints.

## 2 Related Work

Understanding 3D objects and scenes from 2D views is the fundamental task of computer vision. In the early days vision researchers paid close attention to the 2D-to-3D correspondence, but many approaches were line-based and had many difficulties dealing with real-life images. The aspect graph of [1] presents a theory for modeling 3D objects with a set of inter-connected 2D views. This theory has a sound psychological foundation (e.g. [16]) and has been very influential and underlies most approaches to 3D object recognition.

Estimating the 3D pose of objects is a classical problem, and many solutions have been developed using either local features (e.g. [17]) or shape outlines (e.g. [18]), usually assuming perfect knowledge of the object. With the maturation of local feature detection (as in SIFT and its variants), latest progresses on pose estimation have mostly been local-feature based (e.g. [19,20]) and performed fairly well on instances of objects, preferably with texture.

There has been an increasing interest lately in 3D object pose classification, which aims at predicting a discrete set of viewpoints. A variety of approaches have been explored (e.g. silhouette matching [10] or implicit shape models [9] or virtual-training [13]). At the same time, many works on category-level classification also address the issue of multiple views (e.g. [21,14]).

The series of work from Savarese and Fei-Fei [5,6,7] directly address the problem of 3D viewpoint classification at the category and are the most relevant for us. They have developed a number of frameworks for 3D viewpoints, most adopting the strategy of grouping local features into parts and learning about their relations. Similar approaches have been adopted in a number of other works (e.g. [12,22]) that show promising results. The 3DObject dataset of Savarese et al [5] is a standard benchmark for viewpoint classification and has a systematic collection of object views. A number of categories from the PASCAL challenge [8], such as cars, are also annotated with viewpoints. We quantitatively evaluate our approach on these datasets.

The most recent progress in object recognition sees the use of discriminatively trained templates [3,4,23]. These techniques have been shown to perform very well on real-life cluttered images. In particular, the work of [4] presents a way to train mixture-of-components for object detection, and they illustrated the procedure with two components on cars and pedestrians. The context-based discriminative clustering work of [24] is similar in spirit. Our work is based on the mixture-of-HOG approach but focuses on viewpoints instead of categories. We explicitly handle viewpoints and train HOG models with a large number

of viewpoints/components. We also develop approaches for semi-supervised and unsupervised learning of viewpoints, and extend the discrete viewpoint model to the continuous case.

### 3 Discrete Viewpoint Models

In this scheme, given an example object, the models for each category return a confidence score of the object being in that category as well as a *discrete* viewpoint label associated with a canonical pose of that category. In many cases, such poses have semantic meanings, for instance, the frontal/side views of a car. We design each of these models as a mixture of HOG-based templates corresponding to multiple canonical poses of the category. We formulate the score function of example  $x$  as

$$S_{\mathbf{w}}(x) = \max_{v \in \mathcal{V}} \langle w_v, \psi_v(x) \rangle = \max_{v \in \mathcal{V}} w_v^T \psi_v(x) \quad (1)$$

where  $\mathbf{w} = \{w_1, w_2, \dots, w_V\}$  are the learned mixture of templates,  $\mathcal{V} = \{1, 2, \dots, V\}$ ,  $V$  is the number of canonical viewpoints in the model, and  $\psi_v(x)$  is the feature representation of  $x$  under viewpoint label  $v$ . Since the dimensions of templates can be different,  $\psi_v(x)$  is designed to match the dimension of  $w_v$ .

Accordingly, the predicted viewpoint label of  $x$  is

$$\tilde{v}_d(x) = \arg \max_{v \in \mathcal{V}} w_v^T \psi_v(x) \quad (2)$$

where the subscript  $d$  indicates a *discrete* label.

**Features:** We are in favor of HOG-based features because they encode spatial layout of object shape and handle well with intra-class and intra-viewpoint variations. We use the implementation of [4] for feature construction and normalization.

**Detection:** We adopt the standard framework of multi-scale window scanning for localizing objects in the image. The windows whose scores are higher than a learned threshold are picked as candidate detections, and non-max suppression is applied as postprocessing to remove redundant window detections.

#### 3.1 Training

We extend the training algorithm of [4] to cope with viewpoint classification. For each category, we learn a mixture of  $V$ -component templates  $\mathbf{w} = \{w_1, w_2, \dots, w_V\}$  from a set of positive and negative training examples denoted by  $\{x_1, x_2, \dots, x_P\}$  and  $\{z_1, z_2, \dots, z_N\}$ . Our learning framework attempts to “match” every positive example with at least one of these templates, and every negative example with none of the templates. Mathematically, the large margin optimization of this scheme is formulated as

$$(\mathbf{w}^*, \lambda^*) = \arg \min_{\mathbf{w}, \lambda} \sum_{v=1}^V \left\{ \frac{1}{2} \|w_v\|^2 + C_{Neg} \sum_{n=1}^N l(-w_v^T \psi_v(z_n)) + C_{Pos} \sum_{p=1}^P \lambda_v^p \cdot l(w_v^T \psi_v(x_p)) \right\} \quad (3)$$



subject to  $\lambda_v^p \in \{0, 1\}$  and  $\sum_{v=1}^V \lambda_v^p = 1, \forall p = 1, \dots, P$ . Here,  $\lambda$  are binary component labels.  $l(s) = \max(0, 1 - s)$  is the hinge-loss function.  $C_{Pos}$  and  $C_{Neg}$  control the relative weights of the regularization term.

Our training procedure is directly based on [4]: each template  $w_v$  is initialized through a set of positive examples initially labeled as viewpoint  $v$ . In each iteration, all templates are updated simultaneously through data-mining hard negative examples and updating viewpoint labels  $\lambda$  of positive examples.

In [4],  $\lambda$  are considered as latent variables and thus the cost function does not enforce  $\lambda$  to match their true values. Here, we solve a more general problem which includes the scenarios when  $\lambda$  are partially or completely unknown. Furthermore, model initialization in [4] is solely based on aspect ratio; it is not designed for general viewpoint modeling and thus far from optimal for our problem. We will show that a carefully designed initialization is necessary to learn reasonable templates for canonical viewpoints.

Denote  $\{v_d(x_1), v_d(x_2), \dots, v_d(x_P)\}$  as the groundtruth viewpoint labels of the positive examples. In the following, we consider three scenarios, where these labels are completely known, partially known, and unknown. We name them supervised, semi-supervised, and unsupervised cases, respectively.

### 3.2 Supervised Case

In the supervised case, each  $\lambda_v^p = \mathbf{1}[v = v_d(x_p)]$  is fixed. The model is initialized by partitioning the positive examples into groups based on the viewpoint labels and learn one viewpoint template from each group. In the model update step, the optimization is reduced to a linear SVM formulation.

We note that although we do not change component labels during the training process, this is different from training each component independently, as the training process uses a single regularization constraint and enforces the margin on all the clusters simultaneously. This has proved to be critical in learning mixture models that are balanced and accurate for viewpoint prediction.

### 3.3 Semi-supervised Case

In the semi-supervised case, we first build a multi-viewpoint classifier using the positive examples that have known viewpoint labels. In practice, we use the libsvm multi-class classification toolbox [25] on the HOG features. Once the rest of the positive examples are classified, we initialize component templates based on either known or estimated labels. In the model update step, we fix the labels for those who have known viewpoint labels, and allow the others to change.

### 3.4 Unsupervised Case

In the unsupervised case, model initialization is crucial for accurate viewpoint classification, because no explicit constraint in the later stage of optimization is imposed on the viewpoint labels. [4] partitions positive examples into component groups based on a simple aspect ratio criterion. We use a Normalized Cut-based

clustering scheme for initialization. We define an appearance distance between two positive examples  $x_i$  and  $x_j$  as

$$d(x_i, x_j) = \alpha \cdot \chi^2(\psi_0(x_i), \psi_0(x_j)) + (1 - \alpha) \cdot \|\text{Asp}(x_i) - \text{Asp}(x_j)\|_2 \quad (4)$$

where  $\psi_0(x_i)$  is the HOG descriptor of  $x_i$  under a standard template size, and  $\text{Asp}(x_i)$  is the normalized aspect ratio of the bounding box of  $x_i$ . Next, we convert the distances into affinity measurements using the exponential function and obtain the component groups by applying the Normalized Cut [26] algorithm on the resulting affinity matrix. This provides us with relatively even partitionings on the positive examples, which is important for good unsupervised performance.

In the model update step, since Eqn. 3 describes an integer-based non-convex problem ([24], [27]), one tractable solution is to iterate between optimizing  $\mathbf{w}$  given fixed labels  $\lambda$  and optimizing  $\lambda$  given fixed template weights  $\mathbf{w}$ . The former is an SVM and the latter optimization step is simply

$$\lambda_v^p = \mathbf{1}[v = \arg \max_s (w_s^T x_p)] \quad \forall p = 1, \dots, P \quad (5)$$

## 4 Continuous Viewpoint Models

In the continuous viewpoint case, we are interested in estimating the real-valued *continuous* viewpoint angles of an example object in 3D, denoted by  $\theta \in \mathbb{R}^3$ , which uses the angle-axis representation. We assume that the camera projection of the object is orthographic so that given a fixed orientation  $\theta$ , the appearance of the object only changes in scale.

To obtain  $\theta$  for a test object  $x$ , we modify the mixture model in the discrete viewpoint case and reformulate the score function as

$$S_{\mathbf{w}}(x) = \max_{v \in \mathcal{V}, \Delta\theta} f(v, \Delta\theta) = \max_{v \in \mathcal{V}, \Delta\theta} (w_v + g_v \Delta\theta)^T \psi_v(x) - d(\Delta\theta) \quad (6)$$

$$\theta(x) = \theta_{v^*} + \Delta\theta^* \quad (7)$$

where  $\mathbf{w} = \{w_v\}$  and  $\psi_v(x)$  are the same as before.  $g_v$  are the “gradients” of the template  $w_v$  over  $\theta$  at discrete viewpoint  $v$ .  $\Delta\theta$  are the offset viewpoint angles of  $x$  with respect to the canonical viewpoint angles  $\theta_v$ .  $d(\cdot)$  is a quadratic loss function that confines  $\theta(x)$  to be close to  $\theta_v$ . Denote  $\Delta\theta$  by their elements  $[\Delta\theta_1, \Delta\theta_2, \Delta\theta_3]^T$ , then  $d(\Delta\theta) = \sum_{i=1}^3 d_{i1} \Delta\theta_i + d_{i2} \Delta\theta_i^2$ . In Eqn. (7),  $v^*$  and  $\Delta\theta^*$  are obtained when the score function reaches its maximum. The variables  $w_v$ ,  $g_v$ ,  $\theta_v$  and  $d_{i1}$ ,  $d_{i2}$  are learned from training data.

This continuous viewpoint model can be interpreted as follows: we partition the continuous viewpoint space into small chunks where each chunk has a canonical viewpoint. For every viewpoint in the same chunk, we approximate its template as a linear deformation of the canonical template with respect to the difference of viewpoint angles from the canonical angles. We show that in practice, this approximation is reasonable when the chunk size is relatively small, and the model produces viewpoint classification performance superior to a number of baseline methods.

**Detection:** The multi-scale window scanning is again applied for localizing objects in the image. To find optimal  $v$  and  $\Delta\theta$  in Eqn. (6) at a given location, we first maximize  $\Delta\theta$  over any fixed  $v$

$$\frac{\partial f(v, \Delta\theta)}{\partial \Delta\theta_i} = g_v(i)^T \psi_v(x) - d_{i1} - 2d_{i2} \Delta\theta_i = 0 \quad (8)$$

Hence, we obtain

$$\Delta\theta_i(v) = (g_v(i)^T \psi_v(x) - d_{i1}) / 2d_{i2} \quad (9)$$

where  $g_v(i)$  is the  $i$ 'th column of  $g_v$ . Next, we enumerate over the discrete variable  $v$  with  $\Delta\theta_i(v)$  and pick the pair with maximal score  $S_{\mathbf{w}}(x)$ .

#### 4.1 Training

In training, for positive examples  $\{x_1, x_2, \dots, x_P\}$ , their continuous viewpoint groundtruth labels  $\{\theta_1, \theta_2, \dots, \theta_P\}$  are given. Therefore, we rewrite the score function in Eqn. (6) as

$$f(v, \Delta\theta) = (w_v + g_v \Delta\theta)^T \psi_v(x) - d(\Delta\theta) \quad (10)$$

$$= \tilde{w}_v^T \tilde{\psi}_v(x) \quad (11)$$

where

$$\begin{aligned} \tilde{w}_v &= [w_v, g_v(1), g_v(2), g_v(3), d_{11}, d_{12}, d_{21}, d_{22}, d_{31}, d_{32}] \\ \tilde{\psi}_v(x) &= [\psi_v, \Delta\theta_1 \psi_v, \Delta\theta_2 \psi_v, \Delta\theta_3 \psi_v, -\Delta\theta_1, -\Delta\theta_1^2, -\Delta\theta_2, -\Delta\theta_2^2, -\Delta\theta_3, -\Delta\theta_3^2] \end{aligned}$$

If all canonical viewpoint templates  $\theta_v$  are known,  $\psi_v(x)$  are completely observable and we can substitute  $\tilde{w}_v$  and  $\tilde{\psi}_v(x)$  for  $w_v$  and  $\psi_v(x)$  in the training framework of the discrete viewpoint case. Now,  $\theta_v$  are unknown, but we can initialize them from initial partitions of positive data (clustering on  $\theta$ ) and update them in each training iteration based on maximizing the cost function.

## 5 Experimental Evaluation: Discrete Viewpoints

For discrete viewpoint classification, we evaluate our proposed models on two standard and challenging databases: the 3DObject [5] and the VOC2006 cars [8]. The 3DObject dataset consists of 10 categories and 8 discrete viewpoint annotations for each category. We exclude the head and the monitor categories as they are not evaluated in previous work. Quantitative results on viewpoint and category classification are evaluated by means of confusion matrix diagonals, and averaged by 5-fold training/test partitions. On the other hand, the VOC2006 car database consists of 469 car objects that have viewpoint labels (frontal, rear, left and right). In the experiments we only use these labeled images to train mixture viewpoint models, with the standard training/test partition. The detection performance is evaluated through precision-recall curve. For both databases, we try our best to compare with previous works that have the same complete set of evaluations.

In the following sub-sections, we analyze our results in three different levels of supervision on the training data: *supervised*, *semi-supervised*, *unsupervised*.

**Table 1. Supervised Case:** viewpoint and category classification results (quantified by averages of confusion matrix diagonals). For category detection performance on the VOC2006 cars, we compare the precision-recall curves with [7] in Figure 2(d).

Database	3DObject		VOC2006 cars		
Method	[5]	Ours	[6]	[7]	Ours
Viewpoint	57.2%	<b>74.2 ± 0.9%</b>	57.5%	73.0%	<b>85.7%</b>
Category	75.7%	<b>85.3 ± 0.8%</b>	-	-	-

## 5.1 Supervised Case

Table 1 summarizes the viewpoint and category classification results when the viewpoint labels of the positive training data are known. We significantly outperform [5], the state of the art on the 3DObject database, in both viewpoint and category classification. We also show a significantly higher (4-view) viewpoint classification rate on the VOC2006 car database compared to the earlier work of [6] and [7]. Figure 2 shows a close look of our results.

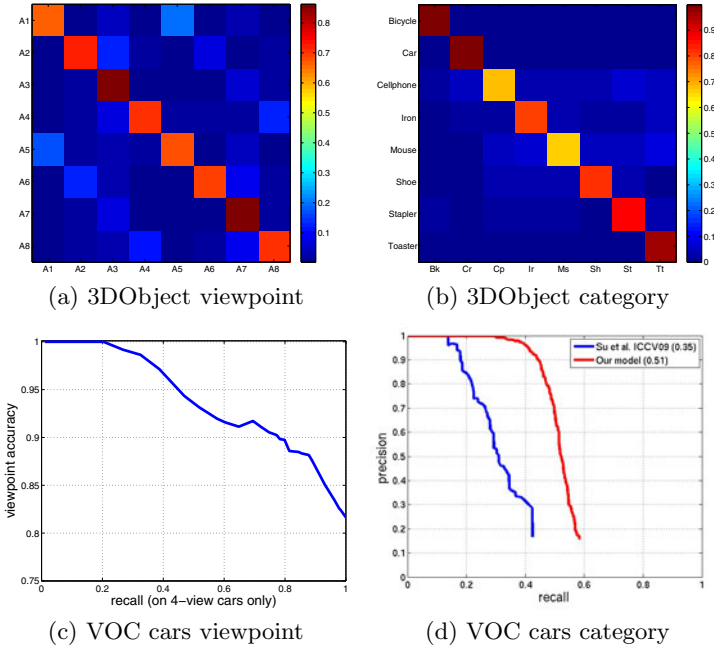
Note that in (a), the main viewpoint confusion pairs in 3DObject are those off by 180 degrees, for example, frontal vs. rear or left vs. right views. Category confusion matrix is shown in (b). (c) illustrates the change of viewpoint classification rate with object recall in VOC2006 cars. The curve suggests that the viewpoint classification accuracy increases with lower recall (and thus higher precision/category detection). (d) compares the precision-recall curves of [7] with ours. Note that even our car mixture model only covers 4 views, it still produces superior performance comparing to [7] in detection.

## 5.2 Semi-supervised Case

In the semi-supervised case, we are interested in knowing how much partial information from positive training data is “sufficient” to build a reasonable viewpoint model. Figure 3 (a, b) illustrate the viewpoint and category classification accuracies with changes in the proportion of training data having discrete viewpoint annotations. Zero proportion means no annotation which corresponds to the unsupervised case, whereas “proportion equals one” is the case of being totally supervised. Note that the accuracy numbers here are evaluated on the whole test set, not the set including only correct category prediction. We notice that even a small proportion (30% in the 3DObject) of annotated data significantly improves the viewpoint classification performance, while the category classification performance remains roughly constant with change of the number of annotated data. (We do not show the curve of category classification on the VOC2006 cars as it is a detection task.)

## 5.3 Unsupervised Case

**Evaluation Methodology.** The upper half of Table 2 compares three model initialization schemes in terms of the viewpoint and category accuracies. We

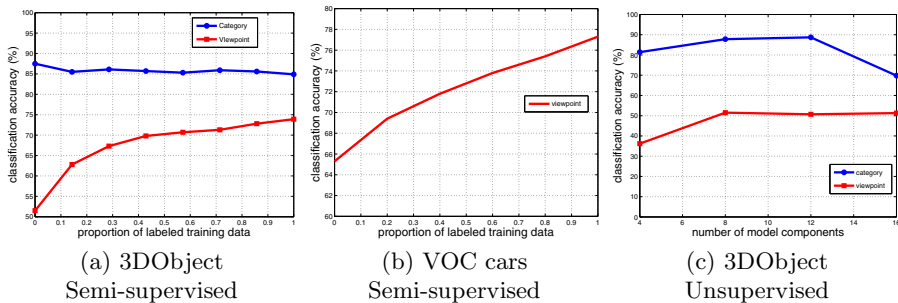


**Fig. 2. Supervised Case:** viewpoint labels are all known for positive examples. (a) (b) Average confusion matrices of the viewpoint and category classifications in the 3DObject. (c) Viewpoint classification accuracy as a function of the object recall in the VOC cars. (d) Precision-recall curves of car detection. Note that our car model only trains on the 4-view cars and tests on the whole test images.

note that our proposed N-cut framework significantly outperformed the aspect ratio criterion by [4] for viewpoint classification. We also compute how far we can reach by computing an “upper bound” performance using the ground truth viewpoint labels of training data in initialization, shown in the third column of the first two databases. We see that the N-cut produces results close to and sometimes even better than the “upper bounds”.

We quantitatively evaluate the quality of viewpoint clustering using the following statistics: *purity*, *normalized mutual information*, *rank index*, and *F measure* [28], shown in the bottom half of Table 2. All measurements of these statistics exhibit consistent behavior as the basic evaluation.

**Number of Model Components.** The number of components  $V$  in the unsupervised model is pre-determined. As a result, we are interested in knowing the impact of this parameter on the viewpoint and category classification performance. Figure 3(c) shows both accuracies with  $V$  on the 3DObject database. Note that for viewpoint classification, the accuracy undoubtedly breaks down when  $V$  is deficient (4) to explain the variety of data in viewpoint (8). It is, however, surprisingly insensitive to  $V$  when it gets large. On the other hand, for



**Fig. 3. Semi-supervised/Unsupervised Cases:** viewpoint and category classification accuracies as a function of either the proportion of positive training data with viewpoint annotations (semi-supervised) or the number of model components/templates (unsupervised). (a): Semi-supervised model on the 3DObject. (b): Semi-supervised model on the VOC2006 cars. For these semi-supervised cases, the category detection performance is robust and largely independent of the availability of viewpoint labels. Viewpoint classification is robust up to about 30% of labeling. (c): Unsupervised model on the 3DObject dataset.

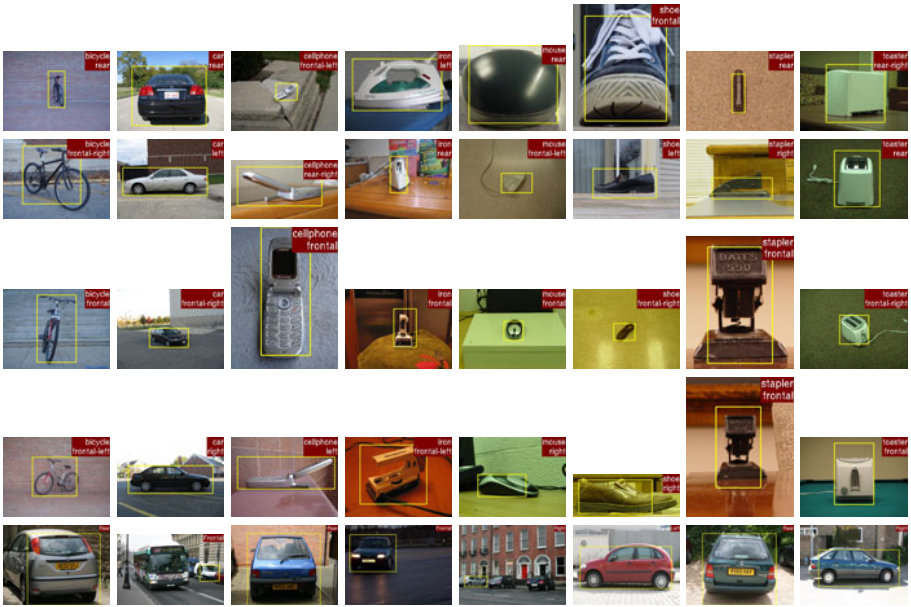
**Table 2. Unsupervised Case:** viewpoint and category classification accuracies as well as four viewpoint clustering measurements [28] on two databases. We show comparison of 3 model initialization schemes ([4], N-cut, and Labels) on the 3DObject and VOC2006 cars. Note that [4] performs poorly in viewpoint classification. The “N-cut”, proposed in this paper where the numbers are bolded, produces significantly better results than [4]. The “Labels” case uses the ground truth viewpoint labels to initialize models, which are considered to produce the “upper-bound” results.

Database	3DObject			VOC2006 cars		
Method	[4]	N-cut	Labels	[4]	N-cut	Labels
Viewpoint	40.2%	<b>51.5%</b>	63.4%	47.0%	<b>65.6%</b>	65.3%
Category	86.5%	<b>87.8%</b>	87.2%	-	-	-
Purity	0.42	<b>0.53</b>	0.65	0.58	<b>0.77</b>	0.76
NMI	0.43	<b>0.55</b>	0.61	0.41	<b>0.52</b>	0.50
Rank Index	0.77	<b>0.83</b>	0.86	0.71	<b>0.80</b>	0.80
F Measure	0.36	<b>0.45</b>	0.54	0.61	<b>0.68</b>	0.67

category classification, the accuracy breaks down when  $V$  is large, and insensitive with small  $V$ .

## 6 Experimental Evaluation: Continuous Viewpoints

For continuous viewpoint estimation, there is no standard benchmark database available, partly because it is considerably harder to establish groundtruth data to cover arbitrary 3D rotations. [19] uses a selected set of translations and rotations for (continuous) pose estimation. [29] does not use groundtruth but



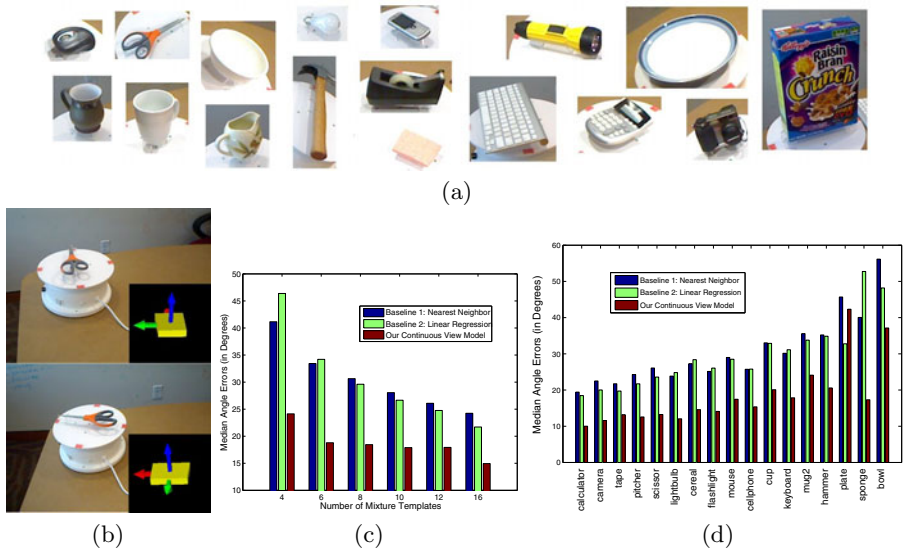
**Fig. 4. Discrete viewpoint classification and category detection results.** The yellow bounding boxes indicate object detection, and the labels on the upper-right corners show the predicted object category and viewpoint. The top 4 rows show results from the 3D object category database, and the bottom row shows results from the PASCAL VOC 2006 car database.

compare results using artificially distorted images. In the case of [30], rotation is limited to in-plane rotation on the ground.

We believe that a good database with full 3D pose groundtruth is crucial for the advances of pose estimation techniques, and we set to collect a 3D pose database using commercially available IMUs: we use the Microstrain 3DM-GX1 sensors and attach it to a PrimeSense video camera. The Microstrain provides gyro-stabilized full 3D orientation at about 80Hz, and the camera records 640x480 frames at 30Hz. The two streams are aligned manually.

We collect a continuous object pose database covering 17 daily objects with a variety of shape, appearance and scale (Fig 5(a)). We put each object on a turning table, let the object turn, while hand-holding the camera/IMU pair and moving it at varying heights and orientations. We typically let each object rotate for 4-5 circles and take about 2K video frames total. In our evaluation experiments, we use all 17 objects and about 1K frames for each object. Frames are evenly and randomly partitioned for training and testing. Object masks are computed from background subtraction and are used to find bounding boxes.

We compare our continuous viewpoint model with two baseline methods. The first one employs a nearest neighbor scheme. Each test example is assigned the same continuous viewpoint label as that of the example’s closest mixture



**Fig. 5. Continuous viewpoint** classification results on the new continuous viewpoint dataset consisting of 17 daily objects (a), covering a wide range of shape and scale. We place objects on a turning table and attach an IMU to a hand-held camera; groundtruth orientations of objects relative to the camera are estimated from both the IMU readings and turning table rotations (b). our discriminatively trained continuous pose model constantly outperforms two baseline methods (assigning to nearest discrete pose, and a linear regression on top of discrete poses). (c) shows the performance comparisons as the number of discrete viewpoints varies in the mixture model. (d) shows the results for each of the 17 objects, with the number of mixture components set to 8. We observe that viewpoint prediction is challenging (and ill-defined for some of the symmetric objects), and our discriminative approach consistently outperforms the baselines for most objects.

template. The second one learns a linear regression model on the responses of all mixture templates to infer viewpoint labels. The comparison of the results is shown in Figure 5(c) where prediction errors are measured by the amount of rotation it takes to go from the predicted pose to the groundtruth pose (in degrees). Because the errors can sometimes be very large due to the symmetry in the object shape and appearance, we use the median angular error as the evaluation metric.

Our proposed continuous viewpoint model constantly outperforms both baselines under different numbers of mixture templates. The errors are reduced as the numbers of templates increase which suggests that a sufficient number of canonical viewpoints is needed to cover the entire viewpoint hemisphere. A closer examination of the per-category performance is shown in Figure 5(d). The errors are in general large for symmetric categories (e.g. plate, bowl) and small for asymmetric ones which meets our intuition. As we see from the examples, the database is challenging: even though the background is simple and so far



instance-based, there is a lot of inherent ambiguity in inferring pose from shape, and the improvement in accuracy using our continuous model is substantial.

## 7 Conclusion

In this work we have applied the discriminative template learning framework for joint category and viewpoint classification. Our main contribution is to show that a mixture-of-templates model discriminatively learned in a detection framework capture the characteristics of different views and can be directly used for viewpoint classification. Our results significantly outperform the state-of-the-art on a number of standard 3D object databases. We have also shown that with a good initialization (e.g. Normalized Cuts and discriminative clustering), we are able to produce meaningful viewpoint clusters and promising classification accuracy with a small amount of training labels.

In addition, we have extended the mixture-of-templates approach to the continuous viewpoint case. We use a linear model to capture local appearance variations at each canonical view, and these models are discriminatively trained as in the discrete case. We have been building up a dataset with continuous viewpoint groundtruth, and our model has shown promising performance comparing to a number of baselines, including discrete nearest neighbor and linear regression.

Although our work is still in a preliminary stage, we believe that our results are very important in proving the use of discriminative learning for viewpoint classification. It is no coincidence that our results outperform the state of the art on 3D object databases. Just as in the category case, discriminative learning addresses the classification problem directly and is very powerful in exploring noisy image data. There are many future opportunities in exploring the synergies between object classification and viewpoint estimation.

## References

1. Koenderink, J., van Doorn, A.: The internal representation of solid shape with respect to vision. *Biological Cybernetics* 32, 211–216 (1979)
2. Lowe, D.: Distinctive image features from scale-invariant keypoints. *Int'l. J. Comp. Vision* 60, 91–110 (2004)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *CVPR* (2005)
4. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. *TPAMI* (2009)
5. Savarese, S., Fei-Fei, L.: 3d generic object categorization, localization and pose estimation. In: *ICCV* (2007)
6. Sun, M., Su, H., Savarese, S., Fei Fei, L.: A multi-view probabilistic model for 3d object classes. In: *CVPR*, pp. 1247–1254 (2009)
7. Su, H., Sun, M., Fei-Fei, L., Savarese, S.: Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In: *ICCV* (2009)

8. Everingham, M., Zisserman, A., Williams, C.K.I., Van Gool, L.: The PASCAL Visual Object Classes Challenge 2006, VOC 2006 Results (2006), <http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf>
9. Arie-Nachmison, M., Basri, R.: Constructing implicit 3d shape models for pose estimation. In: ICCV (2009)
10. Cyr, C., Kimia, B.: A similarity-based aspect-graph approach to 3d object recognition. *Int'l. J. Comp. Vision* 57, 5–22 (2004)
11. Hoiem, D., Rother, C., Winn, J.: 3d layoutcrf 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 (2004)
13. Chiu, H., Kaelbling, L., Lozano-Perez, T.: Virtual-training for multi-view object class recognition. In: CVPR (2007)
14. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *Int'l. J. Comp. Vision* 66, 231–259 (2006)
15. Berg, A., Berg, T., Malik, J.: Shape matching and object recognition using low distortion correspondence. In: CVPR, vol. 1, pp. 26–33 (2005)
16. Bulthoff, H., Edelman, S.: Psychophysical support for a two-dimensional view interpolation theory of object recognition. *PNAS* 89, 60–64 (1992)
17. DeMenthon, D., Davis, L.: Model-based object pose in 25 lines of code. *Int'l. J. Comp. Vision* 15, 123–141 (1995)
18. Lavalley, S., Szeliski, R.: Recovering the position and orientation of free-form objects from image contours using 3d distance maps. *IEEE Trans. PAMI* 17, 378–390 (1995)
19. Collet, A., Berenson, D., Srinivasa, S., Ferguson, D.: Object recognition and full pose registration from a single image for robotic manipulation. In: ICRA (2009)
20. Detry, R., Pugeault, N., Piater, J.: A probabilistic framework for 3D visual object representation. *IEEE Trans. PAMI* 31, 1790–1803 (2009)
21. Thomas, A., Ferrari, V., Leibe, B., Tuytelaars, T., Schiele, B., Gool, L.V.: Towards multi-view object class detection. In: CVPR (2006)
22. Liebelt, J., Schmid, C., Schertler, K.: Viewpoint-independent object class detection using 3d feature maps. In: CVPR (2008)
23. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: NIPS (2009)
24. Lampert, C.: Partitioning of image datasets using discriminative context information. In: CVPR, pp. 1–8 (2008)
25. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
26. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Trans. PAMI* 22, 888–905 (2000)
27. Aioli, F., Sperduti, A.: Multiclass classification with multi-prototype support vector machines. *Journal of Machine Learning Research* (2005)
28. Manning, C.D., Raghavan, P., Schütze, H.: *Introduction to Information Retrieval*. Cambridge University Press, Cambridge (2008)
29. Rosenhahn, B., Brox, T., Weickert, J.: Three-dimensional shape knowledge for joint image segmentation and pose tracking. *Int'l. J. Comp. Vision* 73, 243–262 (2007)
30. Ozuyosal, M., Lepetit, V., Fua, P.: Pose estimation for category specific multiview object localization. In: CVPR (2009)

# Efficient Non-consecutive Feature Tracking for Structure-from-Motion

Guofeng Zhang<sup>1</sup>, Zilong Dong<sup>1</sup>, Jiaya Jia<sup>2</sup>, Tien-Tsin Wong<sup>2</sup>, and Hujun Bao<sup>1</sup>

<sup>1</sup> State Key Lab of CAD&CG, Zhejiang University  
{zhangguofeng,zldong,bao}@cad.zju.edu.cn

<sup>2</sup> The Chinese University of Hong Kong  
{leojia,ttwong}@cse.cuhk.edu.hk

**Abstract.** Structure-from-motion (SfM) is an important computer vision problem and largely relies on the quality of feature tracking. In image sequences, if disjointed tracks caused by objects moving in and out of the view, occasional occlusion, or image noise, are not handled well, the corresponding SfM could be significantly affected. In this paper, we address the non-consecutive feature point tracking problem and propose an effective method to match interrupted tracks. Our framework consists of steps of solving the feature ‘dropout’ problem when indistinctive structures, noise or even large image distortion exist, and of rapidly recognizing and joining common features located in different subsequences. Experimental results on several challenging and large-scale video sets show that our method notably improves SfM.

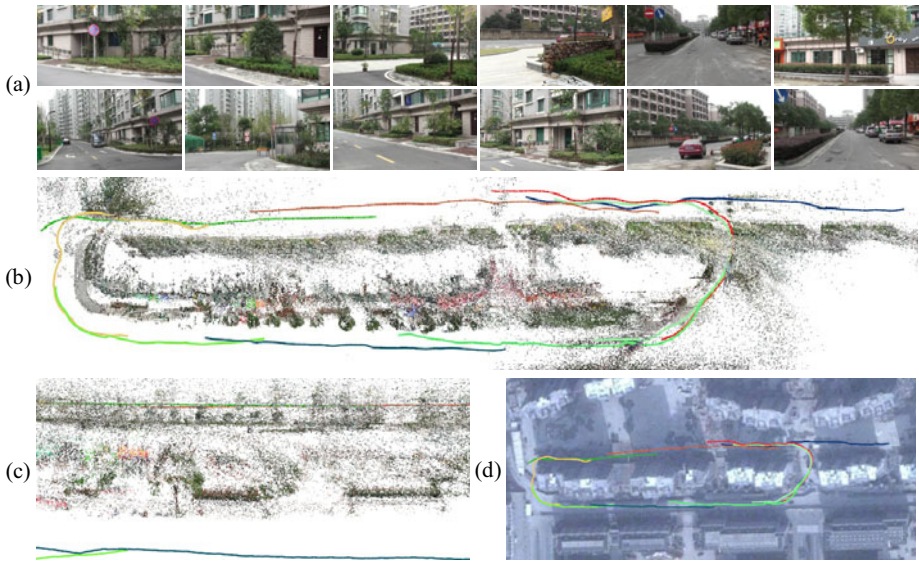
## 1 Introduction

Large-scale 3D reconstruction [1,2,3,4] is a very active research topic and finds many practical applications in, for example, Google Earth and Microsoft Virtual Earth. Recent work essentially relies on the SfM algorithms [5,6,7,4] to automatically estimate 3D features given the input of image or video collections.

Compared to images, videos usually contain denser geometrical and structural information, and are the main source of SfM in the movie and commercial industry. A common strategy for video SfM estimation is by employing feature point tracking [8,9,10,11], which takes care of the temporal relationship among frames. It is also a basic tool for solving a variety of computer vision problems, such as automatic camera tracking, video matching, and object recognition.

In this paper, we discuss two critical and non-trivial problems of feature point tracking, which could seriously handicap SfM especially for large-scale scene modeling, and propose novel methods to address them. One problem is the high vulnerability of feature tracking to object occlusions, illumination change, noise, and large motion, which easily causes occasional feature dropout and distraction. This problem makes developing a robust feature tracking system with the input of long sequences very challenging.

The other problem is the inability of sequential feature tracking to cope with feature matching over non-consecutive subsequences. To our best knowledge, this



**Fig. 1.** “Street” example. (a) The snapshots of the input videos containing around 23,000 frames. (b) With the matched feature tracks, we register many 3D points and camera trajectories in a large 3D system. The camera trajectories are differently colored. (c) Close-up of the recovered trajectories and 3D points. (d) Superimposing the recovered camera trajectories onto a satellite image from Google Earth.

impact has not yet been thoroughly studied in existing literatures. A typical scenario is that the tracked object moves out and then re-enters the field-of-view of the camera. This yields two discontinuous subsequences containing the same object. Although there are common features in the two subsequences, they cannot be matched and included in a single track using conventional tracking methods. Addressing this issue can alleviate the drift problem of SfM, which in turn benefits high-quality 3D reconstruction as demonstrated in our experimental results. A naïve solution to this problem is to exhaustively search all features. But this consumes much unnecessary computation as many temporally far away frames simply share no content.

Our new feature tracking framework efficiently addresses the above problems in two phases, namely *consecutive point tracking* and *non-consecutive track matching*. We demonstrate their significance for SfM estimation using a few challenging videos. *Consecutive point tracking* detects and matches invariant features distributed over consecutive frames. A new two-pass matching strategy is proposed to greatly increase the matching rate of the detected invariant features and extend the lifetime of the tracks. Then in the *non-consecutive track matching* phase, by rapidly computing a matching matrix, a set of disjointed subsequences with overlapping content can be detected. The common feature tracks scattered over these subsequences can also be reliably matched.

Our method can naturally handle feature tracking in multiple videos and register sequences in a large-scale 3D system. Fig. 1 shows a challenging example, which contains 9 videos (about 23,000 frames in total) in a large-scale scene (500 meters long). With our method, a set of long and accurate feature tracks are efficiently obtained. The computation time is only 1.3 seconds per frame with our software implementation (single working thread). Our system also greatly improves SfM by registering videos in a 3D system, as shown in Fig. 1(b). The accuracy of SfM is verified by superimposing the recovered camera trajectories onto a satellite image from Google Earth, as shown in Fig. 1(d). Please refer to our supplementary video 1 for the complete results.

## 2 Related Work

For video tracking, sequential matchers are used for establishing correspondences between consecutive frames. Kanade-Lucas-Tomasi (KLT) tracker [8,9,12] is widely used for small baseline matching. Other advanced methods [11,13,14,15] detect image local features and match them with descriptors.

Both the KLT tracker and invariant feature algorithms depend on modeling feature appearance, and can be distracted by occlusion, similar structures, noise, and image distortion. Generally, sequential matchers cannot match non-consecutive frames under large image transformation. Scale-invariant feature detection and matching algorithms [11,16,2] are effective in recognizing panoramas and in matching wide-baseline images. But they are not easy to be used in consecutive point tracking due primarily to the global indistinctiveness and feature dropout problems in matching, which yield many short tracks.

In addition, invariant features are sensitive to large image distortion. Although variations, such as ASIFT [17], can improve the feature matching performance under substantial viewpoint change, computational overhead significantly increases owing to exhaustive simulation. In this paper, we propose a novel two-pass matching method to solve this problem.

There is work using invariant features for object and location recognition in images/videos [18,19,20,21,22]. These methods typically use the bag-of-words technique to perform global localization and loop-closure detection in an image classification scheme. To reduce the matching ambiguity, they generally suppress indistinctive features. This operation is not suitable for producing long and accurate point tracks.

Engels et al. [23] propose integrating wide-baseline local features with the tracked features to improve SfM. The method creates small and independent submaps over short periods of time and links them together via feature recognition. This approach generally cannot produce many long and accurate point tracks. Only short tracks are found insufficient for drift-free SfM estimation in our experiments. In comparison, our method is effective in high-quality point

---

<sup>1</sup> The supplementary video can be downloaded from <http://www.cad.zju.edu.cn/home/gfzhang/>

track estimation. We also address the ubiquitous nondistinctive feature matching problem in dense frames, and utilize track descriptors, instead of the feature descriptors, to reduce computation redundancy.

### 3 Our Approach

Given a video sequence  $\hat{I}$  with  $n$  frames,  $\hat{I} = \{I_t | t = 1, \dots, n\}$ , the objective of our feature tracking method is to extract and match features in all frames in order to form a set of *feature tracks*. A feature track  $\mathcal{X}$  is defined as a series of feature points in images:  $\mathcal{X} = \{\mathbf{x}_t | t \in f(\mathcal{X})\}$ , where  $f(\mathcal{X})$  denotes the frame set spanned by track  $\mathcal{X}$ . Each invariant feature  $\mathbf{x}_t$  in frame  $t$  is associated with an appearance descriptor  $\mathbf{p}(\mathbf{x}_t)$  [11] and we denote all description vectors in a feature track as  $\mathcal{P}_{\mathcal{X}} = \{\mathbf{p}(\mathbf{x}_t) | t \in f(\mathcal{X})\}$ .

**Table 1.** Overview of Our Method

1. Detect invariant features over the entire sequence.
2. <b>Consecutive point tracking</b> (Section 4):
2.1 Match features between consecutive frames with descriptor comparison.
2.2 Perform the second-pass matching to extend track lifetime.
3. <b>Non-consecutive track matching</b> (Section 5):
3.1 Use hierarchical k-means to cluster the constructed invariant tracks.
3.2 Estimate the matching matrix with the grouped tracks.
3.3 Detect overlapping subsequences and join the matched tracks.

Our method has two main steps, i.e., consecutive point tracking and non-consecutive track matching. The algorithm overview is given in Table 1.

Step 2 in Table 1 suppresses the influence of image noise and distortion in feature tracking, which usually cause spurious feature appearance variation and feature dropout in matching. We locate missing features (as well as the unmatched ones) by a constrained spatial search with planar motion segmentation as described in Section 4.2.

Step 3 is a non-consecutive track matching process. It first uses a hierarchical K-means method to cluster the obtained track descriptors. Based on it, overlapping confidence among non-consecutive frames is measured using a matching matrix, which helps robustly join common features in subsequences. This step is described in Section 5.

### 4 Two-Pass Matching for Consecutive Tracking

In the first place, we use the SIFT algorithm [11] to detect and describe image features. We extract SIFT features from all frames in the input sequence and match them among temporally adjacent frames. The matched features constitute sequential feature tracks. Note that previous KLT methods typically detect

features in the first frame and then track them down consecutively without the invariant feature descriptor constraint. Our method, contrarily, obtains not only a set of feature tracks, but also descriptors to represent tracks, which avail further non-consecutive track matching.

We propose a *two-pass matching* strategy to efficiently reduce false matches caused by structure similarity and feature dropout due to image noise and distortion. The *first-pass matching* is used to obtain high-confidence matches. In the second pass, tracks are extended with planar motion segmentation and constrained spatial search.

#### 4.1 First-Pass Matching by Descriptor Comparison

In this section, we discuss tracking a feature  $\mathcal{X}$  from  $I_t$  to  $I_{t+1}$ . It can be generalized to tracks spanning multiple frames. An invariant feature in  $I_t$  is denoted as  $\mathbf{x}_t$  with descriptor  $\mathbf{p}(\mathbf{x}_t)$ . To determine if there is a corresponding feature  $\mathbf{x}_{t+1}$  with descriptor  $\mathbf{p}(\mathbf{x}_{t+1})$  in  $I_{t+1}$ , we employ the 2NN heuristic proposed by Lowe [11].

Specifically, we search for the two nearest neighboring features of  $\mathbf{x}_t$  in  $I_{t+1}$  with respect to the Euclidean distance of the descriptor vectors and denote them as  $\mathcal{N}_1^{t+1}(\mathbf{x}_t)$  and  $\mathcal{N}_2^{t+1}(\mathbf{x}_t)$ . Their corresponding descriptor vectors are denoted as  $\mathbf{p}(\mathcal{N}_1^{t+1}(\mathbf{x}_t))$  and  $\mathbf{p}(\mathcal{N}_2^{t+1}(\mathbf{x}_t))$  respectively. The matching confidence between  $\mathbf{x}_t$  and  $\mathcal{N}_1^{t+1}(\mathbf{x}_t)$  is defined as

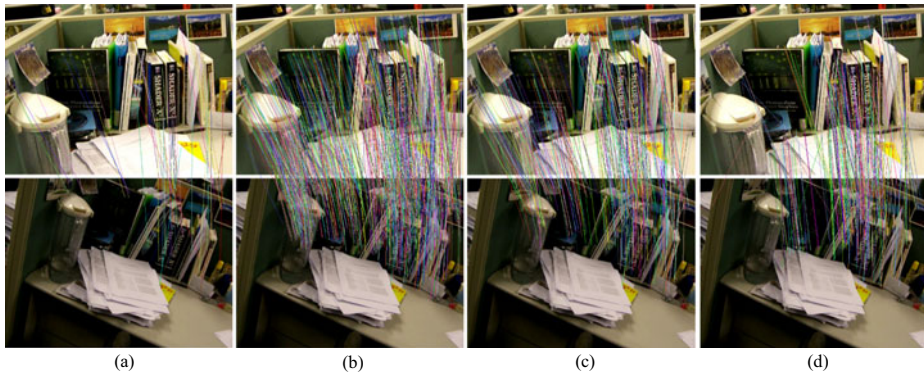
$$c = \frac{\|\mathbf{p}(\mathcal{N}_1^{t+1}(\mathbf{x}_t)) - \mathbf{p}(\mathbf{x}_t)\|}{\|\mathbf{p}(\mathcal{N}_2^{t+1}(\mathbf{x}_t)) - \mathbf{p}(\mathbf{x}_t)\|}, \quad (1)$$

where  $c$  measures the global distinctiveness of one feature  $\mathbf{x}_t$  with respect to the ratio of the smallest feature distance and the second smallest one. If  $c < \varepsilon$ , we assign  $\mathbf{x}_{t+1} = \mathcal{N}_1^{t+1}(\mathbf{x}_t)$  and mark these detected features as *globally distinctive*. In our experiments,  $\varepsilon$  is set to 0.7.

However, this metric is easily interfered by image noise, repeated structures, and image distortion, which make it difficult to find matches for some features even in the adjacent frames. This common problem usually results in breaking a long track into several short ones. One example is shown in Fig. 2. Given an image pair, we detect 1,246 features. Only 50 features can be matched by descriptor comparison, as shown in Fig. 2(a). In the next step, we propose a spatial search method to help identify more matches.

#### 4.2 Second-Pass Matching by Planar Motion Segmentation

With a few high-confidence matches in neighboring frames ( $I_t, I_{t+1}$ ) computed in the first step, we use the RANSAC algorithm [24] to estimate the fundamental matrix  $F_{t,t+1}$  and remove outliers. For those unmatched features, it is possible to search for their correspondences along the conjugate epipolar line  $l_{t,t+1}(\mathbf{x}_t) = F_{t,t+1}\mathbf{x}_t$ . However, if significant image distortion exists, naive



**Fig. 2.** Feature matching comparison. (a) First-pass matching by SIFT descriptor comparison. There are 1,246 features detected; but only 50 matches are found. (b) Second-pass matching by planar motion segmentation. 717 new matches are included; but quite a number of them are outliers. (c) The final result with our outlier rejection. A total of 343 matches are retained. (d) The matching result by ASIFT [17]. 220 matches are found.

---

### Algorithm 1. Second-Pass Matching

---

1. Use the inlier matches to estimate a set of homographies  $\{H_{t,t+1}^k | k = 1, \dots, N\}$  by Algorithm 2 and then use them to obtain a set of rectified images  $\{\hat{I}_t^k | k = 1, \dots, N\}$ .
  2. **for** each unmatched feature  $\mathbf{x}_t$  in  $I_t$  **do**
    - for**  $k = 1, \dots, N$  **do**
      - Find the best match  $\mathbf{x}_{t+1}^k$  by minimizing (2) with  $H_{t,t+1}^k$ .
      - end for**
      - Find the best match  $\mathbf{x}_{t+1}^i$  among  $\{\mathbf{x}_{t+1}^k | k = 1, \dots, N\}$  which minimizes  $S_{t,t+1}^k(\mathbf{x}_t)$ . Further refine  $\mathbf{x}_{t+1}^i$  to  $\mathbf{x}_{t+1}^*$  with the KLT tracking. If  $\|\mathbf{x}_{t+1}^* - \mathbf{x}_{t+1}^i\|$  is large, reject this match.
      - end for**
- 

window-based matching becomes unreliable. Also, an exhaustive search is time-consuming and ambiguous with many potential correspondences. To address these problems, we propose a segmentation-based method (sketched in Algorithm 1) to robustly identify missing matches.

We base our method on the observation that many feature points undergo similar motion. This allows computing inlier matches to estimate a set of homographies  $\{H_{t,t+1}^k | k = 1, \dots, N\}$ , which represent possible local image transformation, as described in Algorithm 2. We then rectify images with their homographies. This scheme is similar to that of [25] where a set of dominant scene planes are extracted to generate a piecewise planar depth map. For an unmatched feature in image  $I_t$ , if its transformation towards  $I_{t+1}$  is coincident with any of these homographies after rectification, a match in  $I_{t+1}$  can possibly be found. Incorrect



**Algorithm 2.** Planar Motion Segmentation

- 
1. Put all matches into a set  $\Omega$ .
  2. For  $k = 1, \dots, N_{\max}$ ,  $\%N_{\max}$  is the maximum number of the homographies.
    - 2.1 Use RANSAC to estimate homography  $H_{t,t+1}^k$  that has the maximum inliers.
    - 2.2 Remove the inliers from  $\Omega$ . If the size of  $\Omega$  is small enough, stop; otherwise, continue.
- 

homographies are unlikely to yield high-confidence matches. To handle illumination change, we estimate the global illumination variation  $L_{t,t+1}$  between images  $I_t$  and  $I_{t+1}$  by computing the average intensity ratio between the matched features.

With the image transformation  $H_{t,t+1}^k$ , we rectify  $I_t$  to  $\hat{I}_t^k$  such that  $\hat{I}_t^k = H_{t,t+1}^k(L_{t,t+1} \cdot I_t)$ . Correspondingly,  $\mathbf{x}_t$  in image  $I_t$  is rectified to  $\hat{\mathbf{x}}_t^k$  where  $\hat{\mathbf{x}}_t^k \sim H_{t,t+1}^k \mathbf{x}_t$  in  $\hat{I}_t^k$ . If  $\hat{\mathbf{x}}_t^k$  largely deviates from the epipolar line (i.e.,  $d(\hat{\mathbf{x}}_t^k, l_{t,t+1}(\mathbf{x}_t)) > 5.0$ ), we reject  $H_{t,t+1}^k$  since it does not describe the motion of  $\mathbf{x}_t$  well. Otherwise, we search for the match along the epipolar line by minimizing the matching cost

$$S_{t,t+1}^k(\mathbf{x}_t) = \min_{\mathbf{x}' \in l_{t,t+1}(\mathbf{x}_t)} \sum_{\mathbf{y} \in W} \|\hat{I}_t^k(\hat{\mathbf{x}}_t^k + \mathbf{y}) - \hat{I}_{t+1}(\mathbf{x}' + \mathbf{y})\|^2, \quad (2)$$

where  $W$  is a  $11 \times 11$  matching window, and  $\mathbf{x}'$  is in the local searching area where  $\|\hat{\mathbf{x}}_t^k - \mathbf{x}'\| < r$  (usually  $r = 15$  in our experiments). The best match is denoted as  $\mathbf{x}_{t+1}^k$ . With the set of homographies  $\{H_{t,t+1}^k | k = 1, \dots, N\}$ , we can find several matches  $\{\mathbf{x}_{t+1}^k | k = 1, \dots, N\}$ . Only the best one  $i = \min_k S_{t,t+1}^k(\mathbf{x}_t)$  is kept.

In case the feature motion cannot be described by all of the homographies or the feature correspondence is indeed missing in the other image, the computed match is actually an outlier. Simply applying threshold  $S_{t,t+1}^i(\mathbf{x}_t) < \tau$  cannot perform satisfactorily, as shown in Fig. 2(b). In addition, the best match may not strictly lie on the epipolar line due to estimation error. We adopt the following procedure to detect outliers.

Our strategy is to relax the epipolar geometry constraint and use the KLT method instead to locally search the best match  $\mathbf{x}_{t+1}^*$ . The intuition is that true correspondence produces the minimum matching cost locally; so searching with and without the epipolar constraint should return the same result. We thus calculate the distance between  $\mathbf{x}_{t+1}^*$  and  $\mathbf{x}_{t+1}^i$ . If  $\|\mathbf{x}_{t+1}^* - \mathbf{x}_{t+1}^i\|$  is large (over 3.0 in our experiments),  $\mathbf{x}_{t+1}^i$  is considered as an outlier; or else,  $\mathbf{x}_{t+1}^*$  is the correspondence of  $\mathbf{x}_t$  and its descriptor is set to that of  $\mathbf{x}_t$ , i.e.  $\mathbf{p}(\mathbf{x}_{t+1}^*) = \mathbf{p}(\mathbf{x}_t)$ .

Applying this criterion effectively rejects most outliers, as shown in Fig. 2(c). Compared to ASIFT [17], our method adaptively estimates a set of dominant homographies, without exhaustively simulating all views. So the computation is much less. Besides, it is hard to apply ASIFT to consecutive point tracking because features, after the simulation of viewpoint change, are no longer the original SIFT ones. Our method has no such problem.

Most of the above steps can be performed hierarchically to yield high efficiency. For a pair of images (resolution  $640 \times 480$ ) with 1,196 features and 4 estimated homographies, the second-pass matching only requires 0.4 second with our software implementation.

The two-pass matching can also produce many long tracks as shown in our supplementary video. Each track has a group of description vectors, denoted as  $\mathcal{P}_{\mathcal{X}} = \{\mathbf{p}(\mathbf{x}_t) | t \in f(\mathcal{X})\}$ . These descriptors must be similar to each other in the same track due to the matching criteria. We compute an average of them and denote it as *track descriptor*  $\mathbf{p}(\mathcal{X})$ . It is used in the following non-consecutive track matching.

## 5 Non-consecutive Track Matching

Given the invariant feature information encoded in tracks, we detect and match features scattered over non-consecutive frames. The following process consists of two main phases, namely *matching matrix estimation* and *non-consecutive track matching*.

### 5.1 Fast Matching Matrix Estimation

To allow non-consecutive track matching, we first estimate a matching matrix for the whole sequence to describe the frame overlapping confidence. Obviously, exhaustive all-to-all frame matching is computationally expensive especially for long sequences. We propose fast estimation of the matching confidence among different frames with regard to the track descriptors.

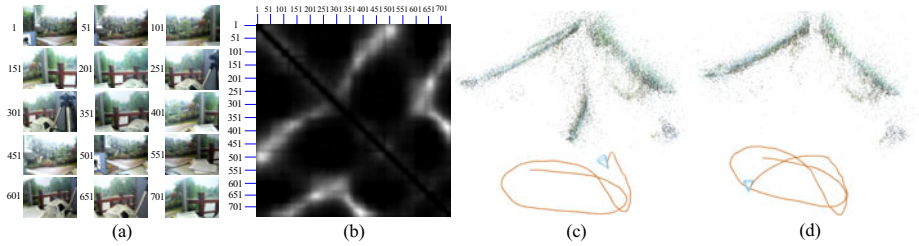
In [26], extracted image descriptors are used to construct a vocabulary tree for fast image indexing. Note that our consecutive point tracking has already clustered matchable features in sequential frames. Instead of locating similar features, we propose constructing a vocabulary tree based on track descriptors for finding similar tracks. This approach can not only significantly reduce the size of the tree, but improve the matching accuracy among non-consecutive frames as well.

We use a hierarchical K-means approach to cluster the track descriptors. The root cluster contains all the descriptors. It is partitioned into  $b$  subgroups by the K-means method. Each sub-cluster consists of the descriptor vectors closest to the center. The same procedure is recursively applied to all subgroups and terminates when the variance of all descriptors in a final (leaf) cluster is less than a threshold. The leaf clusters provide a detailed partition of all tracks. We measure the *overlapping confidence* between any two frames based on the descriptor similarity (depicted in Algorithm 3). The scores are stored in the matching matrix  $M$ , which is with size  $n \times n$ .  $n$  is the number of all frames. The confidence value between images  $I_i$  and  $I_j$  is saved in  $M(i, j)$ .

All elements in  $M$  are first initialized to zeros. In each iteration of Algorithm 3,  $M(i, j)$  is increased by 1 if two features respectively in frames  $i$  and  $j$  are in the same leaf node of the tree. With the objective of non-consecutive frame matching,

**Algorithm 3.** Matching Matrix Estimation

- 
1. Initialize  $M$  as a zero matrix.
  2. For each track cluster  $G_k$  ( $k = 1, \dots, K$ ), %  $K$  is the number of the final clusters.
    - For each track pair  $(\mathcal{X}_u, \mathcal{X}_v)$  in  $G_k$ , if  $f(\mathcal{X}_u) \cap f(\mathcal{X}_v) = \emptyset$ ,
      - For any  $i \in f(\mathcal{X}_u)$  and  $j \in f(\mathcal{X}_v)$ ,
        - $M(i, j) += 1$ ,
        - $M(j, i) += 1$ .
- 



**Fig. 3.** Matching matrix estimation and non-consecutive track matching evaluation. (a) Selected frames from the “wallpaper” sequence. (b) Computed matching matrix that is linearly scaled for visualization. (c) Reconstruction result without non-consecutive track matching. (d) With non-consecutive track matching, the 3D points and camera motion are better estimated. The drift problem is also eliminated, as shown in our supplementary video.

we exclude the cases that two tracks in the same group span common frames (i.e.,  $f(\mathcal{X}_u) \cap f(\mathcal{X}_v) \neq \emptyset$ ).

For acceleration, we only select long tracks in the confidence estimation. In our experiments, for a sequence with 735 frames, the matching matrix estimation only requires 6 seconds, with a total of 22,573 selected feature tracks. Fig. 3(b) visualizes the computed matching matrix from a video, beside which a few selected frames are shown. Bright pixels indicate high confidence. It can be observed that these bright pixels are clustered in different regions in the matching matrix, reflecting the content similarity among subsequences in the input video. The diagonal band has no value because we exclude track self-matching.

## 5.2 Non-consecutive Track Matching

We identify overlapped subsequences by detecting rectangular regions containing the brightest pixels in the matching matrix. Suppose a rectangular region spans  $[u_b, u_e]$  horizontally and  $[v_b, v_e]$  vertically, video subsequences with frame sets  $\phi_1 = \{u_b, \dots, u_e\}$  and  $\phi_2 = \{v_b, \dots, v_e\}$  are correlated.

Since the matching matrix is symmetric, we only consider either the upper or lower triangle. We use the following method to estimate  $[u_b, u_e]$  and  $[v_b, v_e]$ . In the beginning, we search for the element  $M_{ij}$  with the largest similarity value. Then we search for the maximum range  $[u_b, u_e]$  such that  $u_b < i < u_e$  and for any  $t \in [u_b, u_e]$ ,  $M_{tj}/M_{ij} > \delta$ , where  $\delta$  is a threshold.  $[v_b, v_e]$  is computed similarly.

**Algorithm 4.** Track Joining

## 1. Track Matching:

For  $i = u_b, \dots, u_e$ , // one subsequence  $\phi_1$

For  $j = v_b, \dots, v_e$ , // another subsequence  $\phi_2$

Match features in  $I_i$  and  $I_j$  and join their corresponding tracks.

## 2. Outlier Rejection:

For joined tracks  $\mathcal{X}_s$  (in  $\phi_1$ ) and  $\mathcal{X}_t$  (in  $\phi_2$ ), if any of their feature pair  $\{(\mathbf{x}_i^s, \mathbf{x}_j^t) | i \in f(\mathcal{X}_s) \cap \phi_1, j \in f(\mathcal{X}_t) \cap \phi_2\}$  do not satisfy the epipolar geometry constraint, the match is rejected.

Finally, we set  $\phi_1 = \{u_b, \dots, u_e\}$  and  $\phi_2 = \{v_b, \dots, v_e\}$  and set the corresponding elements in the matrix  $M$  to zeros. So in the next round, we again select a new  $M_{ij}$  from the updated matrix  $M$  to detect another subsequence pair for track matching. This process repeats until no high overlapping-confidence frames can be found.

Given the estimated subsequence pair  $(\phi_1, \phi_2)$ , we reliably join tracks scattered over these frame sets (described in Algorithm 4). For each two frames, if their two distinctive features  $\mathbf{x}_i^s$  and  $\mathbf{x}_j^t$ , belonging to  $\mathcal{X}_s$  and  $\mathcal{X}_t$  respectively, are matched using the method described in Section 4.1, we join tracks  $\mathcal{X}_s$  and  $\mathcal{X}_t$  as well. To reject outliers, we apply the geometric constraint to check whether all features in  $\mathcal{X}_s$  and  $\mathcal{X}_t$  satisfy the epipolar geometry constraint, i.e.,  $(\mathbf{x}_i^s, \mathbf{x}_j^t)$  consistent with a fundamental matrix  $F_{ij}$  estimated with the potential matches between frame pair  $(I_i, I_j)$  by the RANSAC algorithm [24]. If the two tracks qualify, they can be safely joined.

The example shown in Fig. 3 demonstrates the effectiveness of our non-consecutive track matching. We perform feature tracking and use the SfM method of [2] to recover camera poses together with sparse 3D points. In the first part of the experiment, we only use sequential tracks to estimate SfM. It is shown in Fig. 3(c) that this scheme produces erroneous camera pose estimate. Then we perform non-consecutive track matching to automatically join common tracks. It improves SfM, as shown in Fig. 3(d). The reconstruction quality can be assessed by inserting a virtual object into the scene, as demonstrated in our supplementary video. When skipping the non-consecutive track matching, the drift problem of the virtual object caused by inaccurate camera pose estimation is severe. In comparison, no such problem is observable after non-consecutive track matching.

### 5.3 Tracks in Multiple Videos

To describe a large-scale scene, multiple videos can generally be obtained from internet or be captured in different geographic regions but generally with overlaps. How to efficiently match multiple videos and register them in a common 3D system was seldom discussed in previous work. In our feature tracking system, this can be naturally accomplished. We first track feature points for each video independently and then detect overlap between each pair of the videos. The

**Table 2.** Running time of a few examples

Datasets	Resolution	Frames	Feature Tracking Time	
			Consecutive	Non-Consecutive
Wallpaper	$640 \times 480$	735	6 minutes	2 minutes
Circle	$960 \times 540$	1,991	30 minutes	12 minutes
Yard	$960 \times 540$	3,201	50 minutes	20 minutes
Street	$960 \times 540$	$\sim 23,000$	6 hours	2 hours

algorithm described in Section 5.1 is used to rapidly estimate the matching matrix such that related subsequences in different videos can be found. Afterwards, we match the common tracks distributed in various subsequences using Algorithm 1. This method quickly yields a matching graph for the collected videos, which finally leads to a robust global 3D registration, as shown in Fig. 1(b).

## 6 Results

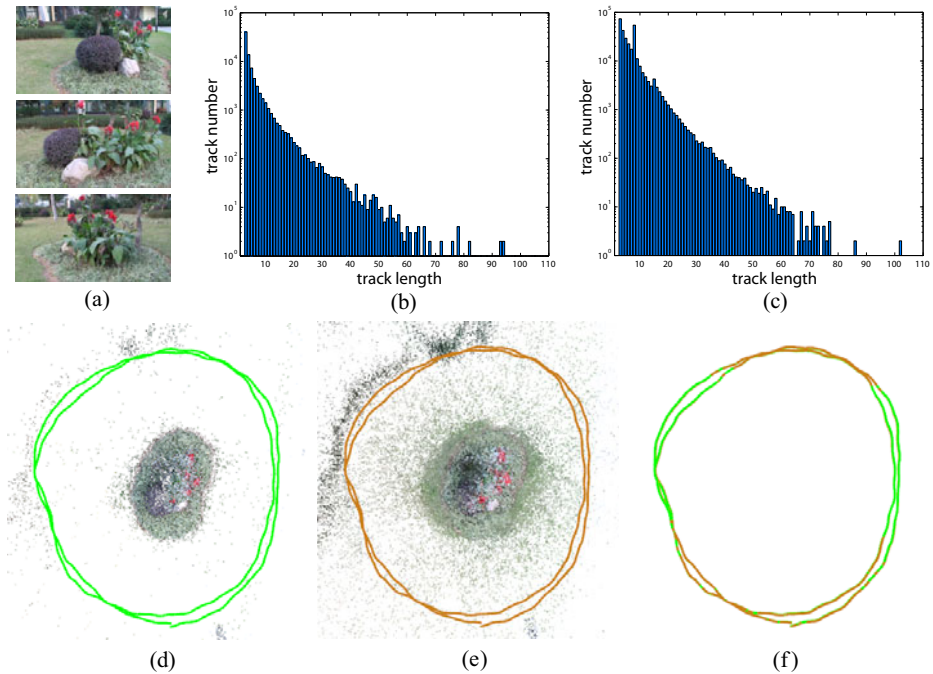
We have evaluated our method on several challenging sequences. All results are generated using a PC with an Intel Core2Duo CPU 2.0GHz and 2GB memory. Running time for feature tracking on the tested data is listed in Table 2.

As our consecutive point tracking can handle wide-baseline images, frame-by-frame tracking is generally not necessary. In our experiments, the system extracts one frame for every 5  $\sim$  10 frames to apply feature tracking. The tracked features are then propagated to other frames by simple KLT sequential tracking. This trick saves a lot of running time and results in feature tracking in a video sequence (1000 features per image and image resolution  $640 \times 480$ ) only taking about 0.5 second per frame with our software implementation (single working thread). The running time of KLT is about 0.4 second per frame. Note that the camera pose estimates from KLT could drift while our method avoids this problem because the computed matches are with very high quality and large quantity.

We compare our method to the brute-force SIFT matching in the Bundler software [27]. The brute-force SIFT matching method does not make use of image ordering. It extracts the SIFT features in all frames and exhaustively compares them. Although a K-d tree is used for matching speedup, the complexity is still quadratic to the number of the processed frames. In contrast, the complexity of our method is almost linear to the frame number.

For the ‘‘circle’’ example with 1991 frames. Performing the brute-force SIFT matching in the whole sequence will take days using our desktop computer. To save time, we pick out one frame for every 5, to compose a new sequence containing only 399 frames. The brute-force SIFT matching spends 187 minutes (6 minutes for SIFT feature extraction) on it, while our method only requires 25 minutes in total. When excluding the SIFT feature extraction time,

<sup>2</sup> We use the CPU implementation downloaded from <http://www.ces.clemson.edu/~stb/klt/>.



**Fig. 4.** Comparison with the brute-force SIFT matching. (a) Three selected frames from the “circle” sequence. (b-c) The track length histograms of the brute-force SIFT matching and our non-consecutive feature tracking, respectively. (d) The SfM result using the feature tracks computed by brute-force SIFT matching. (e) The SfM result using the feature tracks computed by our method. (f) Superimposing the camera trajectory in (d) to (e).

our method is about one order of magnitude faster. Figs. 4(a) and 4(b) show the track length histograms to compare the tracking quality. Our method yields many long feature tracks thanks to the effective two-pass matching and subsequence joining. The SfM results are shown in Figs. 4(d)-(f). The aligned two camera trajectories (shown in Fig. 4(f)) are with average camera position difference 0.000425 (normalized w.r.t. the total length of the camera trajectory).

We tested our method on a challenging large-scale “street” example containing a total of 9 videos, each of which has around 2000 ~ 3000 frames. This example has been shown in Fig. 1. The camera moved along a street and captured several buildings. We first track feature points for each video independently, and then use our non-consecutive track matching algorithm to detect and match common feature tracks across different videos. We perform SfM estimation for each video independently. By aligning the computed 3D points, we register these videos in a 3D system. There are as many as 558,392 estimated 3D points in this example. Superimposing the recovered camera trajectories onto a satellite image shows the high accuracy of the results as all trajectories are on streets and are not drifted.

## 7 Conclusion and Discussion

We have presented a robust and efficient non-consecutive feature tracking system for SfM, which consists of two main steps, i.e., consecutive point tracking and non-consecutive track matching. Different from the typical sequential matcher (e.g. KLT tracker), we use the invariant features and propose a two-pass matching strategy to significantly extend the track lifetime and reduce the feature sensitivity to noise and image distortion. The obtained tracks contain not only a set of 2D image positions, but also descriptors. They avail estimating a matching matrix to detect a set of disjointed subsequences with overlapping views. Our method can also handle tracking and registering multiple videos. Experimental results demonstrate the significance for SfM in middle- and large-scale scenes.

Our method is designed for SfM, and thus consider feature tracking only on rigid (non-deforming) objects in this paper. Part of our future work is to handle deforming or dynamic objects. Besides, although the proposed method is based on the SIFT features, there is no limitation to use other representations, especially in the general two-pass matching process. Further investigation will be conducted.

## Acknowledgements

The work described in this paper was supported by the 973 program of China (No. 2009CB320804), NSF of China (Nos. 60633070 and 60903135), and the Research Grants Council of the Hong Kong Special Administrative Region (Project Nos. 413110 and 417107).

## References

1. Pollefeys, M., Nistér, D., Frahm, J.M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.J., Merrell, P., Salmi, C., Sinha, S.N., Talton, B., Wang, L., Yang, Q., Stewénus, H., Yang, R., Welch, G., Towles, H.: Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision* 78, 143–167 (2008)
2. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. *ACM Trans. Graph.* 25, 835–846 (2006)
3. Li, X., Wu, C., Zach, C., Lazebnik, S., Frahm, J.M.: Modeling and recognition of landmark image collections using iconic scene graphs. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part I. LNCS*, vol. 5302, pp. 427–440. Springer, Heidelberg (2008)
4. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building rome in a day. In: *ICCV*, pp. 72–79 (2009)
5. Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press, Cambridge (2004) ISBN: 0521540518
6. Fitzgibbon, A., Zisserman, A.: Automatic camera tracking. In: *Video Registration*, pp. 18–35 (2003)

7. Zhang, G., Qin, X., Hua, W., Wong, T.T., Heng, P.A., Bao, H.: Robust metric reconstruction from challenging video sequences. In: CVPR (2007)
8. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI, pp. 674–679 (1981)
9. Shi, J., Tomasi, C.: Good features to track. In: CVPR, pp. 593–600 (1994)
10. Georgescu, B., Meer, P.: Point matching under large image deformations and illumination changes. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 674–688 (2004)
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60, 91–110 (2004)
12. Zach, C., Gallup, D., Frahm, J.M.: Fast gain-adaptive klt tracking on the gpu. In: CVPR Workshop on Visual Computer Vision on GPU's (CVGPU) (2008)
13. Lazebnik, S., Schmid, C., Ponce, J.: A sparse texture representation using local affine regions. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 1265–1278 (2005)
14. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 1615–1630 (2005)
15. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. *Image Vision Comput.* 22, 761–767 (2004)
16. Brown, M., Lowe, D.G.: Recognising panoramas. In: ICCV, pp. 1218–1227 (2003)
17. Morel, J.M., Yu, G.: ASIFT: A new framework for fully affine invariant image comparison. *SIAM J. Img. Sci.* 2, 438–469 (2009)
18. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: ICCV, pp. 1470–1477 (2003)
19. Schaffalitzky, F., Zisserman, A.: Automated location matching in movies. *Computer Vision and Image Understanding* 92, 236–264 (2003)
20. Ho, K.L., Newman, P.M.: Detecting loop closure with scene sequences. *International Journal of Computer Vision* 74, 261–286 (2007)
21. Schindler, G., Brown, M., Szeliski, R.: City-scale location recognition. In: CVPR (2007)
22. Irschara, A., Zach, C., Frahm, J.M., Bischof, H.: From structure-from-motion point clouds to fast location recognition. In: CVPR (2009)
23. Engels, C., Fraundorfer, F., Nistér, D.: Integration of tracked and recognized features for locally and globally robust structure from motion. In: VISAPP (Workshop on Robot Perception), pp. 13–22 (2008)
24. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 381–395 (1981)
25. Sinha, S.N., Steedly, D., Szeliski, R.: Piecewise planar stereo for image-based rendering. In: ICCV, pp. 1881–1888 (2009)
26. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR, Washington, DC, USA, pp. 2161–2168. IEEE Computer Society, Los Alamitos (2006)
27. Snavely, N.: Bundler: Structure from motion for unordered image collections, <http://phototour.cs.washington.edu/bundler/>



# *P2II*: A Minimal Solution for Registration of 3D Points to 3D Planes

Srikumar Ramalingam, Yuichi Taguchi, Tim K. Marks, and Oncel Tuzel

Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA

**Abstract.** This paper presents a class of minimal solutions for the 3D-to-3D registration problem in which the sensor data are 3D points and the corresponding object data are 3D planes. In order to compute the 6 degrees-of-freedom transformation between the sensor and the object, we need at least six points on three or more planes. We systematically investigate and develop pose estimation algorithms for several configurations, including all minimal configurations, that arise from the distribution of points on planes. The degenerate configurations are also identified. We point out that many existing and unsolved 2D-to-3D and 3D-to-3D pose estimation algorithms involving points, lines, and planes can be transformed into the problem of registering points to planes. In addition to simulations, we also demonstrate the algorithm's effectiveness in two real-world applications: registration of a robotic arm with an object using a contact sensor, and registration of 3D point clouds that were obtained using multi-view reconstruction of planar city models.

## 1 Introduction and Previous Work

The problem of 3D-to-3D registration is one of the oldest and most fundamental problem in computer vision, photogrammetry, and robotics, with numerous application arise including object recognition, tracking, localization and mapping, augmented reality, and medical image alignment. Recent progress in the availability of 3D sensors at reasonable cost have further accelerated the need for such problems. The registration problem can generally be seen as two subproblems: a correspondence problem, and a problem of pose estimation given the correspondence. Both of these problems are intertwined, and the solution of one depends on the other. This paper addresses the solution to both problems, although the major emphasis is on the second one.

Several 3D-to-3D registration scenarios are possible depending on the representation of the two 3D datasets: 3D points to 3D points, 3D lines to 3D planes, 3D points to 3D planes, etc. [1]. For the registration of 3D points to 3D points, iterative closest point (ICP) and its variants have been the gold standard in the last two decades [2,3]. These algorithms perform very well with a good initialization. Hence for the case of 3D points to 3D points, the main unsolved problem is the initial coarse registration.

The registration of 3D lines to 3D planes and the registration of 3D points *with normals* to 3D planes were considered in [4,5]. (In this paper, we register 3D points without normals to 3D planes.) Recently, there have been several registration algorithms that focus on solving both the correspondence and pose estimation [4,6,7], primarily by casting the correspondence problem as a graph theoretical one. The correspondence

problem maps to a class of NP-hard problems such as minimum vertex cover [8] and maximum clique [9]. In this paper, we address the correspondence problem by formulating it as a maximum clique problem.

The main focus of this paper is on solving for the point-to-plane registration given the correspondence. Despite several existing results in 3D-to-3D registration problems, the registration of points to planes has received very little attention. However, in practice many registration problems can be efficiently solved by formulating them as point-to-plane. Iterative approaches exist for this problem [10,11]. In [1], the authors specifically mention that their algorithms had difficulties with point-to-plane registration and pointed out the need for a minimal solution. The minimal solution developed here provides a clear understanding of degenerate cases of the point-to-plane registration.

The development of minimal solutions in general has been beneficial in several vision problems [11,12,13,14,15,16]. Minimal solutions have proven to be less noise-prone than non-minimal algorithms, and they have been quite useful in practice as hypothesis generators in hypothesize-and-test algorithms such as RANSAC [17]. Our minimal solution for the point-to-plane registration problem also comes with an additional advantage: it dramatically reduces the search space in the correspondence problem.

To validate our theory we show an exhaustive set of simulations and two compelling real-world proof-of-concept experiments: registration of a robotic arm with an object using contact sensor, and registration of 3D point clouds obtained using multi-view reconstruction on 3D planar city models.

*Problem statement:* Our main goal is to compute the pose (3D translation and 3D rotation) of a sensor with respect to an object (or objects) for which a 3D model consisting of a set of planes is already known. The sensor provides the 3D coordinates of a small set of points on the object, measured in the sensor coordinate frame. We are given  $N$  points  $P_1^0, P_2^0, P_3^0, \dots, P_N^0$  from the sensor data and  $M$  planes  $\Pi_1^0, \Pi_2^0, \Pi_3^0, \dots, \Pi_M^0$  from the 3D object. We subdivide the original problem into two sub-problems:

- Compute the correspondences between the 3D points in the sensor data and the planes in the 3D object.
- Given these correspondences, compute the rotation and translation ( $\mathbf{R}_{s2w}, \mathbf{T}_{s2w}$ ) between the sensor and the object. We assume that the object lies in the world reference frame, as shown in Figure 1.

In this paper, we explain our solution to the second problem (pose estimation given the correspondences) in Section 2 before discussing the correspondence problem in Section 3.

## 2 Pose Estimation

In this section, we develop the algorithms for pose estimation given the correspondences between the 3D points and their corresponding planes. Here we assume that the correspondences are already known—a method for computing the correspondences is explained later, in Section 3. We systematically consider several cases in which we know the distribution of the points on the planes (how many points correspond to each plane), developing a customized pose estimation algorithm for each case. We denote

each configuration as  $Points(a_1, a_2, \dots, a_n) \leftrightarrow Planes(n)$ , where  $n = \{3, 4, 5, 6\}$  is the number of distinct planes in which the points lie, and  $a_i$  is the number of points that lie in the  $i$ th plane. The correspondence between a single point and a plane will yield a single coplanarity equation. Since there are 6 unknown degrees of freedom in  $(R_{s2w}, T_{s2w})$ , we need at least 6 point-to-plane correspondences to solve the pose estimation problem. There are also degenerate cases in which 6 correspondences are not sufficient. Although the individual algorithms for the various cases are slightly different, their underlying approach is the same. The algorithms for all cases are derived using the following three steps:

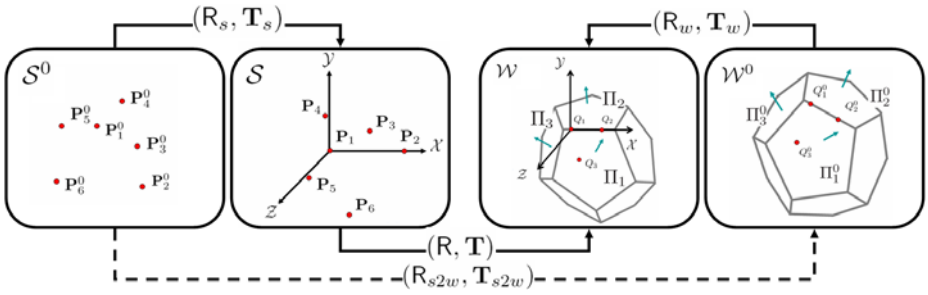
- *The choice of intermediate coordinate frames:* We transform the sensor and the object to intermediate coordinate frames to reduce the degree of the resulting polynomial equations. In addition, if the transformation results in a decrease in the number of degrees of freedom in the pose between the sensor and object, then the rotation  $R$  and the translation  $T$  are expressed using fewer variables.
- *The use of coplanarity constraints:* From the correspondences between the points and planes, we derive a set of coplanarity constraints. Using a linear system involving the derived coplanarity constraints, we express the unknown pose variables in a subspace spanned by one or more vectors.
- *The use of orthonormality constraints:* Finally, we use the appropriate number of orthonormality constraints from the rotation matrix to determine solutions in the subspace just described.

## 2.1 The Choice of Intermediate Coordinate Frames

As shown in Figure 1 we denote the original sensor frame (in which the points reside) and the world reference frame (where the planes reside) by  $S^0$  and  $W^0$ , respectively. Our goal is to compute the transformation  $(R_{s2w}, T_{s2w})$  that transforms the 3D points from the sensor frame  $S^0$  into the world reference frame  $W^0$ . A straightforward application of coplanarity constraints in the case of 6 points would result in 6 linear equations involving 12 variables (the 9 elements of the rotation matrix  $R_{s2w}$  and the 3 elements of the translation vector  $T_{s2w}$ ). To solve for these variables, we would need at least 6 additional equations; these can be 6 quadratic orthonormality constraints. The solution of such a system may eventually result in a polynomial equation of degree  $64 = 2^6$ , which would have 64 solutions (upper bound as per Bezout's theorem), and the computation of such solutions would likely be infeasible for many applications.

To overcome this difficulty, we first transform the sensor and world reference frames  $S^0$  and  $W^0$  to two new intermediate coordinate frames, which we call  $S$  and  $W$ . After this transformation, our goal is to find the remaining transformation  $(R, T)$  between the intermediate reference frames  $S$  and  $W$ . We choose  $S$  and  $W$  so as to minimize the number of variables in  $(R, T)$  that we need to solve for. A similar idea has been used in other problem domains [18]. We now define the transformations from the initial reference frames to the intermediate frames and prove that these transformations are always possible using a constructive argument.

**Transformation from  $S^0$  to  $S$ .** As shown in Figure 1 we represent the  $i$ th point in  $S^0$  using the notation  $P_i^0$  and the same point in  $S$  using  $P_i$ . We define the



**Fig. 1.** The basic idea of coordinate transformation for pose estimation. It is always possible to transform the sensor coordinate system such that a chosen triplet of points  $(P_1, P_2, P_3)$  lie respectively at the origin, on the  $\mathcal{X}$  axis, and on the  $\mathcal{XY}$  plane. On the other hand, the object coordinate frame can always be transformed such that  $\Pi_1$  coincides with the  $\mathcal{XY}$  plane and  $(\Pi_2)$  contains the  $\mathcal{X}$  axis.

transformation  $(R_s, \mathbf{T}_s)$  as the one that results in the points  $(P_1, P_2, P_3)$  satisfying the following conditions: (a)  $P_1$  lies at the origin, (b)  $P_2$  lies on the positive  $\mathcal{X}$  axis, and (c)  $P_3$  lies in the  $\mathcal{XY}$  plane. Note that the points  $P_i^0$  are already given in the problem statement, and the transformation to the points  $P_i$  can be easily computed using the above conditions.

**Transformation from  $\mathcal{W}^0$  to  $\mathcal{W}$ .** We similarly represent the  $i$ th plane in  $\mathcal{W}^0$  using the notation  $\Pi_i^0$  and the same plane in  $\mathcal{W}$  using  $\Pi_i$ . We define the transformation as the one that results in the planes  $\Pi_i$  satisfying the following two conditions: (a)  $\Pi_1$  coincides with the  $\mathcal{XY}$  plane, and (b)  $\Pi_2$  contains the  $\mathcal{X}$  axis.

Assume that  $Q_1^0$  and  $Q_2^0$  are two points on the line of intersection of the two planes  $\Pi_1^0$  and  $\Pi_2^0$ . Let  $Q_3^0$  be any other point on the plane  $\Pi_1^0$ . Let  $Q_1, Q_2,$  and  $Q_3$  denote the same 3D points after the transformation from  $\mathcal{W}^0$  to  $\mathcal{W}$ . The required transformation  $(R_w, \mathbf{T}_w)$  is the one that maps the triplet  $(Q_1^0, Q_2^0, Q_3^0)$  to  $(Q_1, Q_2, Q_3)$ . Note that three points  $Q_i^0$  satisfying the description above can be easily determined from the planes  $\Pi_i^0$ , and the transformation from points  $Q_i^0$  to points  $Q_i$  can be computed in the same way as the transformation described above from points  $P_i^0$  to points  $P_i$ .

We denote the 3D points after the transformation as follows:

$$P_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, P_2 = \begin{pmatrix} X_2 \\ 0 \\ 0 \end{pmatrix}, P_3 = \begin{pmatrix} X_3 \\ Y_3 \\ 0 \end{pmatrix}, \text{ and } P_i = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} \text{ for } i = \{4, 5, 6\}. \quad (1)$$

We write the equations of the planes after the transformation as follows:

$$Z = 0 : \Pi_1 \quad (2)$$

$$B_2 Y + C_2 Z = 0 : \Pi_2 \quad (3)$$

$$A_i X + B_i Y + C_i Z + D_i = 0 : \Pi_i, \text{ for } i = \{3, 4, 5, 6\} \quad (4)$$

**Point-to-plane assignment.** Depending on the particular configuration  $Points(a_1, \dots, a_n) \leftrightarrow Planes(n)$  of the points and planes, we choose which sensor

points correspond to each of  $P_1, P_2, \dots$ , and which object planes correspond to each of  $\Pi_1, \Pi_2, \dots$ , so as to minimize the number of variables in the transformation between the intermediate frames.

In the remainder of this subsection, and in the following subsections [2.2](#) and [2.3](#), we explain the method in the context of a particular example: namely, the configuration  $Points(3, 2, 1) \leftrightarrow Planes(3)$ . For this configuration, we may without loss of generality assume the following correspondences between the points and the planes:

$$\Pi_1 \Leftarrow \{P_1, P_2, P_3\}, \quad \Pi_2 \Leftarrow \{P_4, P_5\}, \quad \Pi_3 \Leftarrow \{P_6\}. \quad (5)$$

As a result of this assignment, the plane corresponding to the three points  $\{P_1, P_2, P_3\}$  and the plane  $\Pi_1$  are both mapped to the  $\mathcal{XY}$  plane. The final rotation ( $\mathbf{R}$ ) and translation ( $\mathbf{T}$ ) between the intermediate sensor coordinate frame  $\mathcal{S}$  and the intermediate object coordinate frame  $\mathcal{W}$  must preserve the coplanarity of these three points and their corresponding plane. Thus, the final transformation can be chosen so as to map all points on the  $\mathcal{XY}$  plane to points on the  $\mathcal{XY}$  plane. In other words, the rotation should be only along the  $\mathcal{Z}$  axis and the translation along the  $\mathcal{X}$  and the  $\mathcal{Y}$  axes. There are two pairs of rotation and translation that satisfy this constraint:

$$\mathbf{R}_1 = \begin{pmatrix} R_{11} & R_{12} & 0 \\ -R_{12} & R_{11} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \mathbf{T}_1 = \begin{pmatrix} T_1 \\ T_2 \\ 0 \end{pmatrix}; \quad \mathbf{R}_2 = \begin{pmatrix} R_{11} & R_{12} & 0 \\ R_{12} & -R_{11} & 0 \\ 0 & 0 & -1 \end{pmatrix}, \mathbf{T}_2 = \begin{pmatrix} T_1 \\ T_2 \\ 0 \end{pmatrix} \quad (6)$$

By choosing assignment [\(5\)](#) and separately formulating  $\mathbf{R}_1$  and  $\mathbf{R}_2$ , we have minimized the number of degrees of freedom to solve for in the transformation between the intermediate frames of reference. Note that  $\mathbf{R}_1$  and  $\mathbf{R}_2$  are related to each other by a  $180^\circ$  rotation about the  $\mathcal{X}$  axis. Below, we explain the algorithm for solving for  $\mathbf{R}_1$  and  $\mathbf{T}_1$ .

### 2.2 The Use of Coplanarity Constraints

To explain our method’s use of coplanarity constraints (and orthonormality constraints), we continue with the example of the specific configuration  $Points(3, 2, 1) \leftrightarrow Planes(3)$ . We know that the points  $P_4$  and  $P_5$  lie on the plane  $\Pi_2$ , whose equation is given by [\(3\)](#). This implies that these points must satisfy the following coplanarity constraints:

$$B_2(-R_{12}X_i + R_{11}Y_i + T_2) + C_2Z_i = 0, \text{ for } i = \{4, 5\} \quad (7)$$

Similarly, the constraint from the third plane  $\Pi_3$  is given below:

$$A_3(R_{11}X_6 + R_{12}Y_6 + T_1) + B_3(-R_{12}X_6 + R_{11}Y_6 + T_2) + C_3Z_6 + D_3 = 0 \quad (8)$$

Using the coplanarity constraints [\(7\)](#), [\(8\)](#), we construct the following linear system:

$$\underbrace{\begin{pmatrix} B_2Y_4 & -B_2X_4 & 0 & B_2 \\ B_2Y_5 & -B_2X_5 & 0 & B_2 \\ A_3X_6 + B_3Y_6 & A_3Y_6 - B_3X_6 & A_3 & B_3 \end{pmatrix}}_{\mathcal{A}} \begin{pmatrix} R_{11} \\ R_{12} \\ T_1 \\ T_2 \end{pmatrix} = \begin{pmatrix} -C_2Z_4 \\ -C_2Z_5 \\ -C_3Z_6 - D_3 \end{pmatrix} \quad (9)$$

The matrix  $\mathcal{A}$  consists of known values and has rank 3. As there are 4 variables in the linear system, we can obtain their solution in a subspace spanned by one vector:

$$(R_{11} \ R_{12} \ T_1 \ T_2)^T = (u_1 \ u_2 \ u_3 \ u_4)^T + l_1 (v_1 \ v_2 \ v_3 \ v_4)^T, \quad (10)$$

where the values  $u_i, v_i$  are known, and  $l_1$  is the only unknown variable.

### 2.3 The Use of Orthonormality Constraints

We can solve for the unknown variable  $l_1$  using a single orthonormality constraint ( $R_{11}^2 + R_{12}^2 = 1$ ) for the rotation variables.

$$(u_1 + l_1 v_1)^2 + (u_2 + l_1 v_2)^2 = 1 \quad (11)$$

By solving the above equation, we obtain two different solutions for  $l_1$ . As a result, we obtain two solutions for the transformation  $(R_1, T_1)$ . Since we can similarly compute two solutions for  $(R_2, T_2)$ , we finally have four solutions for  $(R, T)$ . Using the obtained solutions for  $(R, T)$ , the transformation between the original coordinate frames  $(R_{s2w}, T_{s2w})$  can be easily computed.

**Visualization of the four solutions.** There is a geometric relationship between the multiple solutions obtained for the transformation  $(R, T)$ . For example, in Figure 2(a), we show the four solutions derived above, for a special case in which the 3 planes are orthogonal to each other. All of the solutions satisfy the same set of plane equations, but they exist in different octants. Every solution is just a rotation of another solution about one of the three axes by  $180^\circ$ . If we slightly modify the planes so that they are no longer orthogonal, the different solutions start to drift away from each other.

### 2.4 Other Variants

The example shown above is one of the easiest point-to-plane registration algorithms to derive. Several harder configurations also arise from the distribution of 6 (or more) distinct points on 3 or more planes (see Table 1). We have solved every case using the same intermediate transformation technique described above. All of the different scenarios, the corresponding assignments of points and planes, and the number of solutions are summarized in Table 1.

The key to solving each configuration is to determine a point-to-plane assignment that minimizes the number of variables appearing in the transformation  $(R, T)$  between the intermediate frames. In general, such an optimal assignment can be found by considering different point-to-plane assignments and checking the resulting coplanarity constraint equations for the 6 points and their corresponding planes. For example, in the configuration  $Points(3, 2, 1) \leftrightarrow Planes(3)$ , the point-to-plane assignments given in (5) minimize the number of unknowns in the equations (6) for  $(R, T)$ . Please see the Supplementary Materials for details of various configurations summarized in Table 1.

**Special cases.** If the points lie on the boundaries of the planes (i.e., every point lies on two planes), then 3 points are sufficient to compute the pose. A careful analysis shows that this problem is nothing but a generalized 3-point pose estimation problem [20].

**Table 1.** Point-to-plane configurations and their solutions

Each row of the table presents a different configuration, in which  $n$  denotes the number of distinct planes and each  $a_i$  refers to the number of points that lie in the  $i$ th plane. The first two rows show the degenerate cases for which there is an insufficient number of points or planes. The next four rows consider non-minimal solutions using more than 6 points. The remaining rows show several minimal configurations (each using exactly 6 points). The number of solutions is given, followed by the average number of real (non-imaginary) solutions in parentheses based on 1000 computations from the simulation described in Section 5. Processing time was measured using a MATLAB implementation on a 2.66 GHz PC; the symbol † indicates the use of Groebner basis methods [19]. The Supplementary Materials explain the derivations of the various configurations.

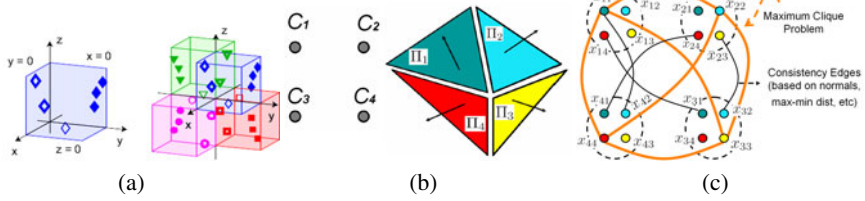
$n$	$(a_1, \dots, a_n)$	Assignment	# of Solutions	Process time (msec)
$< 3$	-	-	degenerate	-
$n$	$\sum a_i < 6$	-	degenerate	-
3	(3,3,3)	$\Pi_1 \Leftarrow \{P_1, P_2, P_3\}, \Pi_2 \Leftarrow \{P_4, P_5, P_6\}, \Pi_3 \Leftarrow \{P_7, P_8, P_9\}$	2 (2)	5
3	(3,3,2)	$\Pi_1 \Leftarrow \{P_1, P_2, P_3\}, \Pi_2 \Leftarrow \{P_4, P_5, P_6\}, \Pi_3 \Leftarrow \{P_7, P_8\}$	2 (2)	5
3	(3,3,1)	$\Pi_1 \Leftarrow \{P_1, P_2, P_3\}, \Pi_2 \Leftarrow \{P_4, P_5, P_6\}, \Pi_3 \Leftarrow \{P_7\}$	2 (2)	5
3	(3,2,2)	$\Pi_1 \Leftarrow \{P_1, P_2, P_3\}, \Pi_2 \Leftarrow \{P_4, P_5\}, \Pi_3 \Leftarrow \{P_6, P_7\}$	2 (2)	5
3	(4,1,1)	-	degenerate	-
3	(3,2,1)	$\Pi_1 \Leftarrow \{P_1, P_2, P_3\}, \Pi_2 \Leftarrow \{P_4, P_5\}, \Pi_3 \Leftarrow \{P_6\}$	4 (4)	6
3	(2,2,2)	$\Pi_1 \Leftarrow \{P_5, P_6\}, \Pi_2 \Leftarrow \{P_3, P_4\}, \Pi_3 \Leftarrow \{P_1, P_2\}$	8 (4.4)	140 <sup>†</sup>
4	(3,1,1,1)	$\Pi_1 \Leftarrow \{P_1, P_2, P_3\}, \Pi_2 \Leftarrow \{P_4\}, \Pi_3 \Leftarrow \{P_5\}, \Pi_4 \Leftarrow \{P_6\}$	4 (2.8)	6
4	(2,2,1,1)	$\Pi_1 \Leftarrow \{P_5, P_6\}, \Pi_2 \Leftarrow \{P_3, P_4\}, \Pi_3 \Leftarrow \{P_2\}, \Pi_4 \Leftarrow \{P_1\}$	8 (3.6)	140 <sup>†</sup>
5	(2,1,1,1,1)	$\Pi_1 \Leftarrow \{P_5, P_6\}, \Pi_i \Leftarrow \{P_{6-i}\}, i = \{3, 4, 5\}$	16 (5.8)	410 <sup>†</sup>
6	(1,1,1,1,1,1)	$\Pi_i \Leftarrow \{P_{6-i+1}\}, i = \{1, 2, 3, 4, 5, 6\}$	16 (5.8)	1200 <sup>†</sup>

**Degenerate cases.** Table 1 includes several degenerate cases based on the number of points and planes. In addition, degeneracies can occur based on the geometry of the planes. In the case of 3 planes, if the  $3 \times 3$  matrix consisting of all three normals has rank less than 3 (e.g., if two of the three planes are parallel), it is a degenerate configuration.

### 3 The Correspondence Problem

In the previous section, we assumed that the point-to-plane correspondences were known. In this section, we briefly describe a method to compute these correspondences. The basic idea of the correspondence problem and the geometrical constraints involved in identifying feasible correspondences are explained in detail in [5] using an interpretation tree approach. The same problem can also be formulated as graph-theoretical problems such as independent set, vertex cover and maximum clique [5,8,9].

Our goal in this section is to compute all of the feasible mappings (possible assignments) between the 3D points in the sensor domain and planes in the object. Feasible mappings refer to correspondences that satisfy the many geometrical constraints arising from the angles between the normals, pairwise distances, etc. [5]. Although such constraints do not always guarantee the correctness of the mappings, a wrong correspondence seldom exists satisfying all the constraints. In addition, since we use them in hypothesize-and-test algorithms such as RANSAC, outliers can be detected and removed.



**Fig. 2.** (a) *Right* Visualization of 4 solutions for the points lying on 3 orthogonal planes. *Left*: Correct solution. (b) The problem of finding correspondences between clusters of points  $C_i$  and planes  $\Pi_j$ . (c) This can be formulated as a maximum clique problem. Each node  $x_{ij}$  in this graph represents a mapping between cluster  $C_i$  and plane  $\Pi_j$ . An edge between two nodes is a consistency edge, signifying that both of these mappings can occur simultaneously without conflicting with the three constraints given in [5].

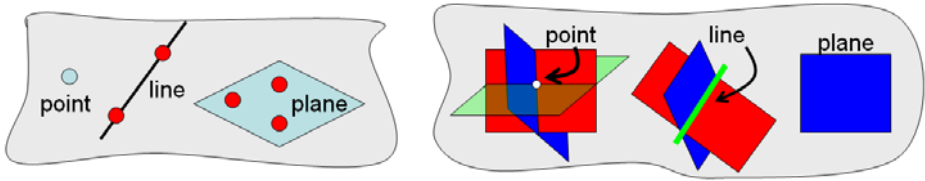
In what follows, we briefly explain our approach using the maximum clique problem formulation. First, we cluster the points from the sensor into several planes, denoting the  $i$ th cluster as  $C_i$ . Note that each cluster may contain multiple points or even just a single point. As shown in Figure 2(b), our goal is to map these clusters to the corresponding planes  $\Pi_j$  in the object. In order to do this, we construct a graph as shown in Figure 2(c). Every node in this graph  $x_{ij}$  represents a mapping between the cluster  $C_i$  (from the sensor) and the plane  $\Pi_j$  (from the object). An edge between  $x_{ij}$  and  $x_{kl}$  is referred to as a consistency edge that signifies that both these mappings can occur simultaneously without conflicting with the three constraints given in [5]. The feasible correspondences between points and planes can be obtained by finding the maximum clique in the graph. A maximum clique for a graph refers to the largest subset of nodes in which each pair of nodes in the subset is connected by an edge. In the graph we constructed, finding a maximum clique provides us a set of mappings in which all possible pairwise consistencies are satisfied.

Several techniques can be used to solve these NP-hard problems [8,7]. Since we use minimal approaches for our applications, we are not interested in the correspondences for all of the points in the registration problem. Instead, we are concerned with identifying a small number of point-to-plane correspondences (sufficient to resolve issues from degeneracies and outliers). In fact, one of the main advantages of the proposed minimal solution is that it only requires correspondences for a small number of points. This enabled us to use a simple tree-based search for finding the maximum cliques in the real-world experiments described in Section 5.

## 4 A General Framework for Pose Estimation

We briefly sketch a unified pose estimation framework for most 2D-to-3D and 3D-to-3D registrations by first transforming the given problem to a point-to-plane registration problem. Several 2D-to-3D pose estimation algorithms have been proposed in the literature [6,18,10,12,15,4,20]. All of these pose estimation algorithms involve the registration of one set of geometrical entities (points, lines, or planes) to another. For example,





**Fig. 3.** A general framework to transform a given registration problem to a point-to-plane problem. *Left:* In the sensor data, we transform all geometrical entities (points, lines and planes) to points. A point is preserved as a point. In the case of lines and planes we sample two and three arbitrary points, respectively. *Right:* In the object data, we convert all geometrical entities to planes. A plane is preserved as a plane. Points and lines are parameterized using 3-plane and 2-plane representations, as shown.

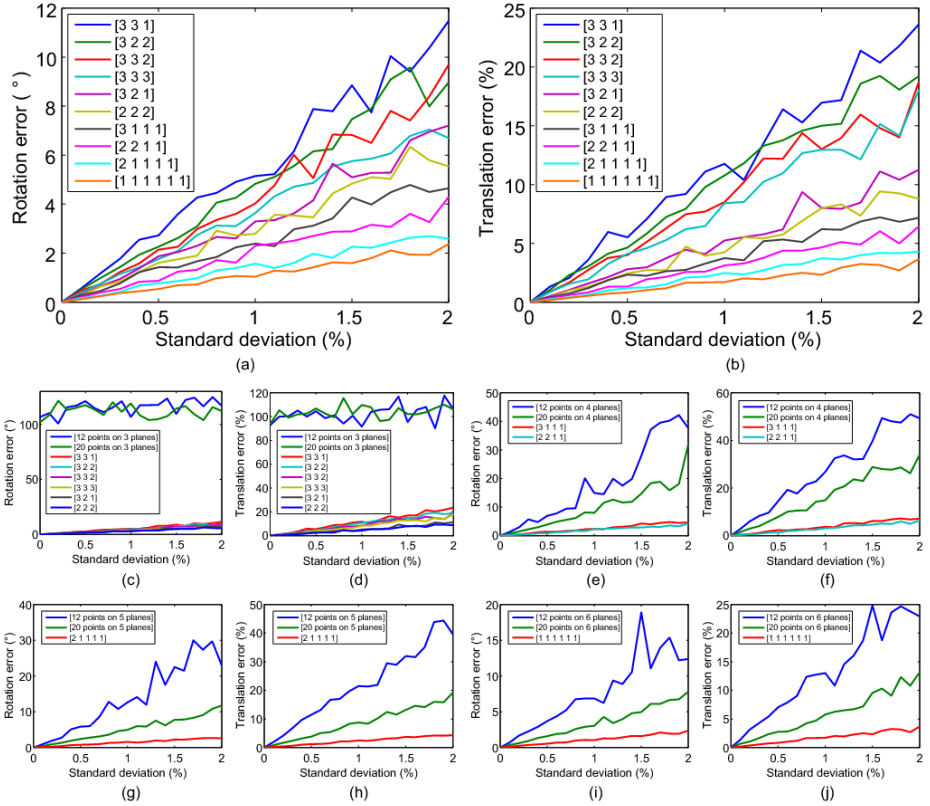
in the case of generalized pose estimation, we register three 3D points to the corresponding non-parametric projection rays from the cameras to compute the pose of the object with respect to the camera [20]. In the case of 2D-to-3D pose estimation using three lines, we can look at this problem as a registration of three interpretation planes (each formed by two projection rays corresponding to a single line) on three lines [18]. In the case of 3D-to-3D line-to-plane registration, we register lines from the sensor data to planes from the object [4]. In the case of 3D-to-3D point-to-point registration, we register points from sensor data to points in the object [6]. One could also propose registration algorithm involving mixture of geometrical entities and thereby we could have more than 20 2D-to-3D and 3D-to-3D registration scenarios. We emphasize that any of these pose estimation algorithms involving any combination of geometrical entities to any other combination could be transformed to a point-to-plane registration algorithm and solved using the following simple algorithm.

1. In the sensor data, we transform all the geometrical entities (points, lines and planes) to points. This is done using 2-point and 3-point representation of lines and planes respectively as shown in Figure 3.
2. In the object data, we transform all the geometrical entities to planes. This is done by 3-plane and 2-plane representations for points and lines, respectively. Note that the 3 planes passing through a point need not be orthogonal. Similarly, we use 2 non-orthogonal planes to represent a line. The appropriate choice of these planes plays a crucial role in obtaining an efficient pose estimation algorithm.
3. After these transformations, we can use our point-to-plane registration algorithm.

Details of the proposed generalized framework are given in the Supplementary Materials with examples on several registration problems.

## 5 Experimental Results

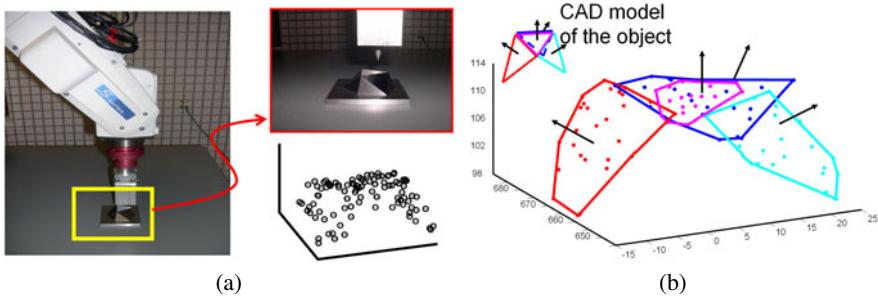
**Simulations.** We analyzed the performance of our minimal solutions in simulations by generating 32 random planes inside a cube of side length 100 units. We randomly sampled 320 points on these planes within the cube. A test set was created by transforming



**Fig. 4.** Rotation and translation error for simulation data as a function of the level of noise in the test set. The noise standard deviation is expressed as a percentage of the size of the object. The legends list the configurations in order of decreasing error. (a,b) Results from our algorithm for all non-degenerate configurations shown in Table I. Note that minimal solutions using 6 points provide lower errors than non-minimal solutions, and solutions for configurations with larger number of planes have lower errors. (b–j) Our minimal solutions compared to least square methods (using 12 and 20 points) for the same number of planes  $n$ : (c,d)  $n = 3$ , (e,f)  $n = 4$ , (g,h)  $n = 5$ , and (i,j)  $n = 6$ . Note that in the 3-plane case (b), least square methods completely fail due to rank degeneracy.

all 320 points using a ground-truth rotation and translation, then adding Gaussian noise to each point.

We randomly selected  $k$  points from the test set according to the point-to-plane configuration of the algorithm, then computed the rotation and translation using the points and the corresponding planes. The estimated transformation was then evaluated by using it to transform the other  $320 - k$  points and computing the mean point-to-plane distance between the transformed points and their correct corresponding planes. Each trial consists of generating a test set, then repeating the selection of  $k$  points and transformation estimation 100 times for this test set. Of the resulting 100 transformations,

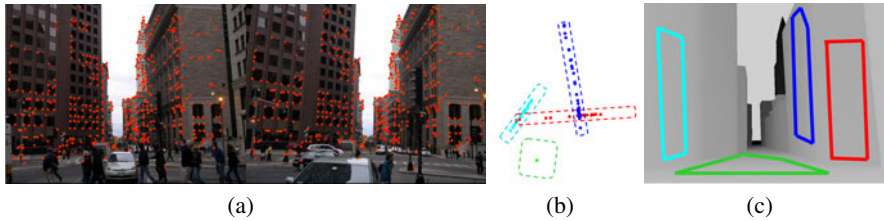


**Fig. 5.** Real-world experiment with a 6-degrees-of-freedom robotic arm. (a) 3D contact position data were collected for 100 points on the surface using a built-in contact detection function and built-in encoders of the robotic arm. (b) Plane fitting of the 3D points and the correspondences of the points to the planes in the CAD model using the method of Section 3.

the solution for the trial is the one transformation that provides the minimum mean distance.

Figure 4 plots errors in estimated rotation and translation with varying noise levels. For each configuration, the errors plotted are the average of 100 trials. For each number of planes ( $n = 3, 4, 5, 6$ ), we compare our minimal solutions for every possible configuration of 6 points (as well as the non-minimal configurations for 3 planes that were included in Table 1) to a least-squares solution for the same number of planes using 12 or 20 points without orthonormality constraints. In all cases, our minimal solutions yield smaller errors than the least squares method. Note that the least squares method completely fails in the case of three planes. Thus, our transformation is useful not only for the minimal configurations but also in non-minimal configurations such as  $(3, 3, 3)$ .

**Contact Sensor.** The first experiment, shown in Figure 5, was conducted using a 6-degree-of-freedom robotic arm with a built-in contact detection function. We used as the target object a partial surface of an icosahedron, of which four of the 20 faces are measurable, as shown in Figure 5. The robot automatically measured 100 points (contact positions) on the surface; each point was measured by first moving the probe to a random  $x, y$  position and then moving down towards the surface (in the negative  $z$  direction) until it sensed a contact. We clustered the points using a simple RANSAC-based plane fitting algorithm. There were four main clusters corresponding to the four planes of the icosahedron used in the experiment. Next, the method described in Section 3 was used to find the correspondences between these clusters and the planes in the 3D model. Given these correspondences, we applied our point-to-plane algorithm using several of the minimal 3-plane and 4-plane configurations. As in the simulations, we repeated the following process to determine the solution: randomly selecting  $k$  points, solving for the transformation, and evaluating the mean distance of the transformed remaining points to the 3D model. The final point-to-plane distance error for all of the inliers was about 3% of the overall size of the scene. The least squares method failed completely for the 3-plane case (similar to the results shown in Figure 4). In the 4-plane case, the least-squares error was about 10 times larger than the error of the minimal solutions.



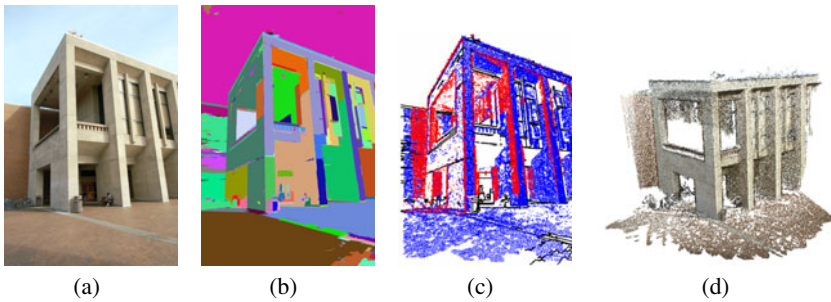
**Fig. 6.** (a) An input stereo pair of photos taken in Boston’s financial district, overlaid with the points that we matched and reconstructed in 3D. (b) We identify four clusters in the reconstructed 3D points (a single point and three planar clouds of points) using a plane-fitting algorithm. (c) The four planes in the 3D city model corresponding to the identified clusters shown in (b).

**Registration of 3D point clouds to polyhedral architectural models.** Given a plane-approximated coarse 3D model of the city of Boston obtained from a commercial website (<http://www.3dcadbrowser.com/>), we performed localization within the map using a pair of images of a scene in Boston’s financial district. To obtain 3D points from the image pair, we matched Harris features and applied standard structure-from-motion algorithms.

Using a RANSAC-based plane fitting algorithm, we fit planes to the reconstructed 3D points. We computed 3 planes from the reconstructed points as shown in Figure 6. A coarse initialization is manually provided and the nearest planes in the 3D model are identified. All of the planes shown in Figure 6(c) (more than 10 planes) were used from the 3D model of Boston. Using the method described in Section 3, we obtained the correspondences between four clusters (a single point and three planar clouds of points) and four planes in the 3D model. The plane corresponding to the ground had only one 3D point due to occlusion from pedestrians and cars. (Note that it was important to have at least one point on the ground in order to determine the vertical translation.) Applying our minimal algorithms for the 4-planes case yielded results with an error of just 0.05% of the overall size of the scene.

Our point-to-plane registration algorithm can also be used for merging partial reconstructions obtained from multi-view reconstruction techniques [22][23], as shown in Figure 7. In order to obtain a 3D model from 30 images, we subdivide the images into two clusters of 15 images each. We reconstruct 3D point clouds from each image cluster and use the superpixel segmentation of a common image to register them. The 3D points from the first cluster are reprojected onto the superpixel image and used to compute the plane parameters for each superpixel. (We eliminate superpixels with insufficient or non-planar points.) The superpixel segmentation of the common image gives us the correspondences between the points in the second cluster and the planes obtained from the first cluster. We obtain the 3D registration using a RANSAC framework, in which we select three or more non-degenerate planes (See section 2.4) and the corresponding minimum number of points.

Previous work merging partial 3D models obtained multi-view 3D reconstruction has used non-minimal iterative approaches [24]. However, initializing with a minimal solution, such as the one described here, may be critical for noisy 3D data. In addition,



**Fig. 7.** Registering two point clouds, each generated by applying multi-view reconstruction techniques to 15 images. **(a)** One of the images used in 3D reconstruction. **(b)** superpixel segmentation of the image shown in (a). **(c)** The 3D points from the first (blue) and second (red) clouds are reprojected onto the superpixel image. The points from the first point cloud are used to compute the superpixel plane parameters, while the second point cloud is preserved as points. The correspondence between the points from the second cloud and the planes obtained from the first cloud are determined by the underlying superpixel. **(d)** 3D model after merging the two partial reconstructions from the two clusters. **[Best viewed in color]**

there are two general advantages of point-to-plane rather than point-to-point registration: (1) accuracy [25], (2) compact representation of the 3D models (about a million 3D points are represented using few hundred superpixel planes).

## 6 Discussion

The development of minimal algorithms for registering 3D points to 3D planes provides opportunities for efficient and robust algorithms with wide applicability in computer vision and robotics. Since 3D sensors typically do not perceive the boundaries of objects in the same way as 2D sensors, an algorithm that can work with points on the surfaces, rather than surface boundaries, is essential. In textureless 3D models, for example, it is easier to obtain point-to-plane correspondences than point-to-point and line-to-line correspondences.

**Acknowledgments.** We would like to thank Jay Thornton, Keisuke Kojima, John Barnwell, and Haruhisa Okuda for their valuable feedback, help and support.

## References

1. Olsson, C., Kahl, F., Oskarsson, M.: The registration problem revisited: Optimal solutions from points, lines and planes. In: CVPR, vol. 1, pp. 1206–1213 (2006)
2. Besl, P., McKay, N.: A method for registration of 3D shapes. PAMI (1992)
3. Fitzgibbon, A.: Robust registration of 2d and 3d point sets. In: Image and Vision Computing (2003)
4. Chen, H.: Pose determination from line-to-plane correspondences: Existence condition and closed-form solutions. PAMI 13, 530–541 (1991)

5. Grimson, W., Lozano-Prez, T.: Model-based recognition and localization from sparse range or tactile data. MIT AI Lab, A.I. Memo 738 (1983)
6. Horn, B.: Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society A* 4, 629–642 (1987)
7. Li, H., Hartley, R.: The 3D-3D registration problem revisited. In: ICCV, pp. 1–8 (2007)
8. Enqvist, O., Josephson, K., Kahl, F.: Optimal correspondences from pairwise constraints. In: ICCV (2009)
9. Tu, P., Saxena, T., Hartley, R.: Recognizing objects using color-annotated adjacency graphs. In: Forsyth, D., Mundy, J.L., Di Gesù, V., Cipolla, R. (eds.) *Shape, Contour, and Grouping 1999*. LNCS, vol. 1681, p. 246. Springer, Heidelberg (1999)
10. Chen, Y., Medioni, G.: Object modeling by registration of multiple range images. In: ICRA, vol. 3, pp. 2724–2729 (1991)
11. Kukulova, Z., Pajdla, T.: A minimal solution to the autocalibration of radial distortion. In: CVPR (2007)
12. Gao, X., Hou, X., Tang, J., Cheng, H.: Complete solution classification for the perspective-three-point problem. *PAMI* 25, 930–943 (2003)
13. Stewenius, H., Nister, D., Kahl, F., Schaffalitzky, F.: A minimal solution for relative pose with unknown focal length. In: CVPR (2005)
14. Stewenius, H., Nister, D., Oskarsson, M., Astrom, K.: Solutions to minimal generalized relative pose problems. In: OMNIVIS (2005)
15. Geyer, C., Stewenius, H.: A nine-point algorithm for estimating para-catadioptric fundamental matrices. In: CVPR (2007)
16. Li, H., Hartley, R.: A non-iterative method for correcting lens distortion from nine-point correspondences. In: OMNIVIS (2005)
17. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24, 381–395 (1981)
18. Dhome, M., Richetin, M., Lapresté, J.T., Rives, G.: Determination of the attitude of 3-D objects from a single perspective view. *PAMI* 11, 1265–1278 (1989)
19. Kukulova, Z., Bujnak, M., Pajdla, T.: Automatic generator of minimal problem solvers. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part III*. LNCS, vol. 5304, pp. 302–315. Springer, Heidelberg (2008)
20. Nistér, D.: A minimal solution to the generalized 3-point pose problem. In: CVPR (2004)
21. Haralick, R., Lee, C., Ottenberg, K., Nolle, M.: Review and analysis of solutions of the three point perspective pose estimation problem. *IJCV* (1994)
22. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multi-view stereopsis. *PAMI* (2009)
23. Furukawa, Y., Ponce, J.: Patch-based multi-view stereo software (2000), <http://grail.cs.washington.edu/software/pmvs>
24. Ramalingam, S., Lodha, S.: Adaptive enhancement of 3d scenes using hierarchical registration of texture-mapped 3d models. In: 3DIM (2003)
25. Rusinkiewicz, S., Levoy, M.: Efficient variants of the icp algorithm. In: 3DIM (2001)

# Boosting Chamfer Matching by Learning Chamfer Distance Normalization

Tianyang Ma, Xingwei Yang, and Longin Jan Latecki

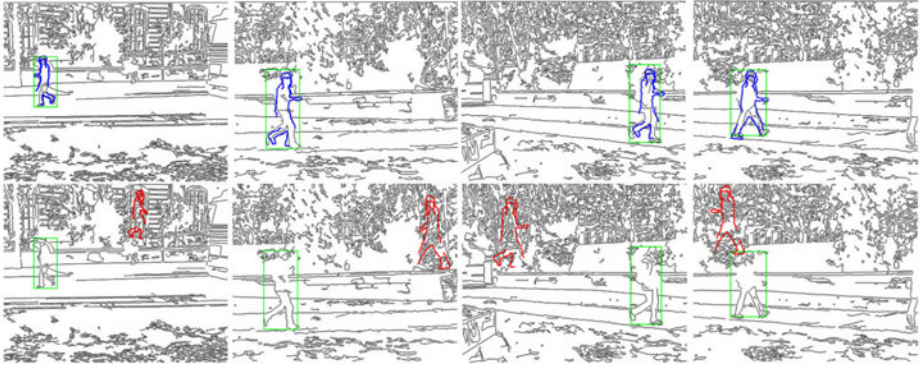
Dept. of Computer and Information Sciences, Temple University, Philadelphia  
{tianyang.ma,xingwei,latecki}@temple.edu

**Abstract.** We propose a novel technique that significantly improves the performance of oriented chamfer matching on images with cluttered background. Different to other matching methods, which only measures how well a template fits to an edge map, we evaluate the score of the template in comparison to auxiliary contours, which we call normalizers. We utilize AdaBoost to learn a Normalized Oriented Chamfer Distance (NOCD). Our experimental results demonstrate that it boosts the detection rate of the oriented chamfer distance. The simplicity and ease of training of NOCD on a small number of training samples promise that it can replace chamfer distance and oriented chamfer distance in any template matching application.

## 1 Introduction

Chamfer matching has been widely used for edge based object detection and recognition in computer vision. However, its performance is seriously limited in cluttered images. One of the main drawbacks of chamfer matching is the fact that a given template often fits better to a cluttered background than to the location of a true target object. Oriented chamfer matching (OCD) [17] adds orientation information, which significantly improves the performance of chamfer matching, but the problem still remains, as illustrated in Fig. 1. The proposed approach provides a solution to this problem by comparing the matching score of the template to normalizers, which are curve segments of varying but simple shape. There are two key properties of the normalizers. (1) If the target template matches well to a cluttered background, then very likely some of the normalizers match well too. (2) If the template matches well to a true object location, it is very unlikely for any normalizer to match well. Consequently, the normalized oriented chamfer distance (NOCD) significantly improves the discriminative power of OCD. Some examples are shown in Fig. 1.

Since it is hard if not impossible to satisfy (1) and (2) with a finite set of normalizers for a given set of target templates, we treat normalized chamfer distances as weak classifiers and employ AdaBoost to learn their weights. The weights provide a soft way of selecting adequate normalizers for a given template. As our experimental results demonstrate, AdaBoost is able to learn the normalizer weights on a small set of training images, which makes the proposed



**Fig. 1.** Example detection results on 250 test images from TU Darmstadt Pedestrian Dataset. The first row shows the detection results of the proposed NOCD, while the second row shows oriented chamfer matching results. The green rectangle denotes the ground truth bounding box.

approach suitable for all practical applications currently based on (oriented) chamfer matching.

The paper is structured as follows. In Section 3, we review basic definitions of chamfer distance and oriented chamfer distance. The new concept of distance normalization is introduced in Section 4 and AdaBoost learning of their weights is described in Section 5. Section 6 describes a simple framework for object detection. Finally, Section 7 introduces our set of normalizers. The performance of our method is evaluated and compared to OCD in Section 8.

## 2 Related Work

There is a large number of applications of chamfer matching in computer vision and in medical image analysis. Chamfer distance was first introduced by Barrow et al. [2] in 1977 with a goal of matching two collections of contour fragments. Until today chamfer matching is widely used in object detection and classification task due to its tolerance to misalignment in position, scale and rotation. Borgfors [16] introduced a modified chamfer matching method called hierarchical chamfer matching, which could be regarded as a coarse-to-fine process by matching edge points using a resolution pyramid of the image. This method focuses on alleviating the computational load for chamfer matching. Meanwhile, chamfer matching meets the real-time system requirement due to fast implementations of distance transforms. Gavrilu and Munder [3] performed template matching based on chamfer distance transform as a core technique to construct a real-time detection system of pedestrians.

Leibe et al. [4] used chamfer matching to detect pedestrian in crowded scenes, and combined segmentation as a verification to prevent the false alarms that



mostly lie in the cluttered background. Stenger et al. [6] introduced a template hierarchy which is formed by bottom-up clustering based on the chamfer distance. In [7], Opelt et al. used chamfer distance to score each boundary fragment for selection of candidate contour fragments. Opelt et al. also compared each boundary fragment from each category to all existing alphabet entries using chamfer distance in [8]. Other methods that utilize chamfer distance as shape similarity metric include [9,13,20]. Chamfer distance plays also an important role in medical image analysis, e.g., [10,11,12].

However, methods that utilize chamfer distance to measure the similarity between the template and edge maps suffer from mismatching to the cluttered background. It is generally agreed that main negative effect of using chamfer distance is the potential risk of increasing false alarms occurring in background with high level of clutter noise. Thayananthan et al. [14] compared the localization performance of chamfer matching and shape context [15], and concluded that chamfer matching is more robust in clutter than shape context matching even though most failure cases in chamfer matching are still due to false positive matches.

Recently, Shotton et al. [17] proposed an oriented chamfer distance (OCD) that exploits edge orientation information in the form of edge gradients. OCD linearly combines chamfer distance and orientation difference between template points and their closest matches, which leads to reduction of mismatching cases to the noisy background. Trinh and Kimia [25] proposed Contour Chamfer Matching (CCM) to improve OCD. In this method, based on the observation that the accidental alignment between a contour and the image edges always forms a zig-zagging contour, after finding the corresponding points in edge map, another orientation for edge points is computed based on the new generated curve, and an additional term which is the difference in tangent direction is taken into account when computing the Contour Chamfer Distance.

Since proposed method is not designed specifically for oriented chamfer distance, it could be also used to boost the performance of any distance metric that aims to capture edge support for a model. In particular, it would be possible to apply the proposed method to Hausdorff distance and oriented Hausdorff distance proposed in [26,27], which is also widely used in computer vision applications. However, in [17] experimental evidence is provide that OCD has better performance than Hausdorff distance.

### 3 Oriented Chamfer Distance (OCD)

In this section we define chamfer distance and oriented chamfer distance (OCD), which is a simple linear combination between distance and orientation terms.

**Chamfer Distance.** Chamfer distance was first proposed in [2] as an evaluation of 2D asymmetric distance between two set of edge points. It is tolerant to slight shape distortion caused by shift in location, scale and rotation. Given a template  $T$  positioned at location  $x$  in an image  $I$  and a binary edge map  $E$  of the image  $I$ , the basic form of chamfer distance is calculated as

$$d_{cham}^{(T,E)}(x) = \frac{1}{|T|} \sum_{x_t \in T} \min_{x_e \in E} \|(x_t + x) - x_e\|_2, \quad (1)$$

where  $\|\cdot\|_2$  is  $l_2$  norm and  $|T|$  denotes number of points in template  $T$ . Chamfer distance can be efficiently computed as:

$$d_{cham}^{(T,E)}(x) = \frac{1}{|T|} \sum_{x_t \in T} DT_E(x_t + x), \quad (2)$$

where  $DT_E$  is a distance transform defined for every image point  $x \in I$  as

$$DT_E(x) = \min_{x_e \in E} \|x - x_e\|_2. \quad (3)$$

Meanwhile, in practice, distance transform is truncated to a constant  $\tau$  [17]:

$$DT_E^\tau(x) = \min(DT_E(x), \tau) \quad (4)$$

This reduces the negative effective due to missing edges in  $E$ , and allows normalization to a standard range  $[0, 1]$ :

$$d_{cham,\tau}^{(T,E)}(x) = \frac{1}{\tau|T|} \sum_{x_t \in T} DT_E^\tau(x_t + x). \quad (5)$$

**Oriented Chamfer Distance (OCD).** Shotton et al. [17] proposed an improved chamfer distance called oriented chamfer distance (OCD), which adds additional robustness by exploiting edge orientation information. To define it, we first need a notation of an argument of a distance transform (ADT) that gives the locations of a closest point.

$$ADT_E(x) = \arg \min_{x_e \in E} \|x - x_e\|_2. \quad (6)$$

To evaluate a mismatch in orientation, the difference in tangent directions is computed

$$d_{orient}^{(T,E)}(x) = \frac{2}{\pi|T|} \sum_{x_t \in T} |\phi(x_t) - \phi(ADT_E(x_t + x))|, \quad (7)$$

where  $\phi(x)$  denotes tangent direction at point  $x$  and ranges between zero and  $\pi$ .  $|\phi(x_1) - \phi(x_2)|$  gives the smallest circular difference between  $\phi(x_1)$  and  $\phi(x_2)$ . Using a simple linear combination between the distance and orientation terms, oriented chamfer distance is defined as

$$OCD_\lambda^{(T,E)}(x) = (1 - \lambda) \cdot d_{cham,\tau}^{(T,E)}(x) + \lambda \cdot d_{orient}^{(T,E)}(x). \quad (8)$$

For clarity, we will omit  $E$  and  $\lambda$  below when possible, and use  $OCD(T, x) = OCD_\lambda^{(T,E)}(x)$  to represent the oriented chamfer distance of template  $T$  at location  $x \in I$ .

## 4 Normalization of Oriented Chamfer Distance

Although oriented chamfer matching adds orientation term to avoid mismatching, cluttered background still may match much better to the template than the real object contours. The reason is that cluttered background offers a large variety of edge orientations, consequently, any shape has a large probability of a good oriented chamfer score. This suggests that we need to compare the score of the target template with scores of some random shapes. If both have good OCD score at a given location, then the template match is most likely to be accidental. Based on this insight, we introduce a normalizer as an auxiliary, random shape to evaluate how well the template matches to the edge map at a certain location. For a target template  $T$ , we propose to generate  $K$  normalizers, denoted by  $\mathcal{N} = \{\eta_k \mid k = 1, \dots, K\}$ . A procedure to generate normalizers is described in Section 7. Instead of only calculating  $OCD(T, x)$  at each location  $x$ , we also compute  $OCD(\eta_k, x)$ , and compare the ratios

$$R_k(T, x) = \frac{OCD(T, x)}{OCD(\eta_k, x)}. \quad (9)$$

We call  $R_k(T, x)$  a **normalized score**.

Now we provide some details about the role of normalizers in improving chamfer score. The analysis is divided into three qualitative cases that illustrate an intended correct behavior of the normalizers. In practice, not all normalizers will behave in this way, which is addressed in Section 5.

**Case 1:** At a correct location containing a target object in a given image,  $OCD(T, x)$  is small and  $OCD(\eta_k, x)$  is large, so that  $OCD(T, x) < OCD(\eta_k, x)$ . Consequently,  $R_k(T, x)$  will become comparatively smaller than  $OCD(T, x)$ , which better indicates a correct match.

**Case 2:** In a cluttered area in which the target object is not present, both  $OCD(T, x)$  and  $OCD(\eta_k, x)$  are small, but  $OCD(T, x) > OCD(\eta_k, x)$ , so  $R_k(T, x)$  will become comparatively larger than  $OCD(T, x)$ , which better indicates a wrong match.

**Case 3:** In an area that is neither cluttered nor contains the target object, both  $OCD(T, x)$  and  $OCD(\eta_k, x)$  are large, but  $OCD(T, x) > OCD(\eta_k, x)$ , so  $R_k(T, x)$  will become comparatively larger than  $OCD(T, x)$ , which better indicates a wrong match.

Cases 1 to 3 clearly demonstrate that normalizers increase the discriminate power of OCD. However, they are based on an assumption that we have an ideal set of normalizers  $\{\eta_k \mid k = 1, \dots, K\}$  behaving as described in cases 1 to 3. Even though it may not be possible to find normalizers satisfying cases 1 to 3 for a given template  $T$ , we propose to utilize machine learning methods to learn which normalizers yield correct scores  $R_k(T, x)$  for a given template  $T$ . For a given set of candidate normalizers, we use AdaBoost in Section 5 to learn the weights of normalized scores  $R_k(T, x)$ . Thus, we treat each normalized score as a weak classifier. The weights provide a soft selection of a set of normalizers with our intuition being that this selection best approximates the behavior described in cases 1 to 3.

## 5 Learning Normalized OCD with AdaBoost

The standard AdaBoost [18] allows us to select a set of normalizers by assigning weights to their normalized scores and to combine them as a weighted linear combination, which yields a more robust matching score. Given is a set of training images with positive and negative examples, i.e, a set of bounding boxes containing the target object and a set of bounding boxes without the target object. AdaBoost automatically learns the weight for each weak learner and combine them to form a strong learner [21][22]. We use the ratios  $R_k(T, x)$  as weak learners for  $k = 1, \dots, K$ . To be precise, a weak learner is defined as

$$h_k(T, x) = \begin{cases} 1 & \text{for } R_k(T, x) < th_k \\ 0 & \text{for } otherwise. \end{cases} \tag{10}$$

In each iteration  $1, \dots, K$ , we search for a weak learner with the best detection performance on the training set. During the search, the optimal threshold  $th_k$  for each weak learner is chosen to minimize the misclassification error (ME). At each iteration of AdaBoost, each training example carries a classification weight. ME is defined as the sum of the classification weights of misclassified training examples (both positives and negatives). As the output we obtain a strong learner

$$H(T, x) = \sum_{k=1}^K w_k \cdot h_k(T, x) \tag{11}$$

In the AdaBoost terminology, the value of the strong learner indicates how likely a given image location  $x$  belongs to the class of template  $T$ . The larger the value the most likely this is the case. We propose to replace the oriented chamfer distance of  $T$  with the value of  $H(T, x)$ . We define a **Normalized Oriented Chamfer Distance** as  $NOCD(T, x) = H(T, x)$ . While OCD is a distance in that the smaller is OCD value the better, NOCD is a similarity measure, i.e., the larger the NOCD value, the most likely the target object is present at location  $x$ .

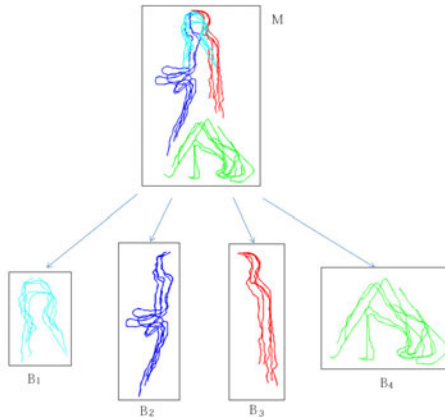
We use a simple strategy to select training examples for AdaBoost. Given is a set of training images with ground truth bounding boxes enclosing target objects. For each training image we select only 5 positive and 5 negative examples. As 5 positive examples we randomly select 5 locations in a small neighborhood around the ground truth locations. We select as negative examples 5 locations  $x$  with locally smallest oriented chamfer distance  $OCD(T, x)$  such that the area of the intersection of the bounding box centered at  $x$  with any ground truth bounding box is less than 50%.

## 6 Object Detection with NOCD

In order to be able to evaluate the performance of NOCD, we describe a very simple approach for object detection in this section. We keep it simple to allow

for clear comparison to OCD. However, we use a flexible shape model in our approach in order to be able to evaluate the performance of the proposed *NOCD* on state-of-the-art test datasets.

Our flexible object model is denoted as  $\mathcal{M} = \{B_i \mid i = 1, \dots, N\}$ , where  $B_i$  is a part bundle composed of contour parts describing the same location on the contour of a given shape class, e.g., human head or arm, and  $N$  is the number of bundles in model  $\mathcal{M}$ . Contour parts from bundle  $B_i$  are represented by  $c_{ij}$ , and hence  $B_i = \{c_{ij} \mid j = 1, \dots, M_i\}$ . Since every part bundle  $B_i$  describes a specific part of an object, we assume that  $B_i \cap B_j = \emptyset$  if  $i \neq j$ . Fig. 2 shows an example of human model, here  $N = 4$  and  $M_i = 5$  for  $i = 1, 2, 3, 4$ . Our model was manually constructed. Thus, our model contains the total of 20 contour parts  $c_{ij}$ . Each part  $c_{ij}$  is treated as template  $T$ , and  $NOCD(c_{ij}, x)$  is learned as describe in Section 5.



**Fig. 2.** Human model  $\mathcal{M}$  composed of 4 part bundles  $B_1, B_2, B_3, B_4$  representing head, front, back, and leg parts, respectively. Each bundle has 5 contour parts.

For an input image  $I$ , we first use Canny edge detector to compute the edge map  $E$ . For each location  $x$  in  $I$ , we use  $NOCD(c_{ij}, x)$  to represent the normalized oriented chamfer distance of model contour part  $c_{ij}$  placed at point  $x$ . With a simple but efficient sum-max framework, the model fit at point  $x \in I$  is defined as:

$$S_I(\mathcal{M}, x) = \sum_{i=1}^N \max_{c_{ij} \in B_i} NOCD(c_{ij}, x) . \quad (12)$$

Thus, we select from each bundle  $B_i$  the part with the largest NOCD score and sum the maximal scores over the bundles in the shape model  $\mathcal{M}$ . Using sliding window we calculate  $S_I(\mathcal{M}, x)$  at each point  $x \in I$ . We define the model fit score as

$$S_I(\mathcal{M}) = \max_{x \in I} S_I(\mathcal{M}, x) \quad (13)$$

and the detection center point as point  $x^* \in I$  as

$$x^* = \arg \max_{x \in I} S_I(\mathcal{M}, x) \quad (14)$$

The detection results for *OCD* follow the same framework, but with max replaced with min in the above formulas.

## 7 Normalizers

It remains to describe how we select a set of normalizers  $\{\eta_k \mid k = 1, \dots, K\}$ . We first observe that a good normalizer should be more likely to match to noise than a given contour part. This implies that a normalizer should have a significantly simpler shape than the contour parts of a target shape model. We also want that a normalizer should be less likely to match to a true object edges in an image than a given contour part. Consequently, normalizers should not be similar to any contour parts in our shape models.



**Fig. 3. Basic normalizers.** Our set of basic normalizers contains 11 simple shapes.

We satisfy both constrains by first generating a small set of simple geometric curves that are treated as a basic structuring elements to generate a set of normalizers. A set of 11 basic shapes that we have selected is shown in Fig. 3. They form the first 11 elements of our set of normalizers  $\mathcal{N} = \{\eta_k \mid k = 1, \dots, K\}$ . We obtain further normalizers by pairwise combining the 11 structuring elements, where the combination is simply a union of their aligned images. Since the normalizer combination is symmetric and we only combine different structuring elements, we obtain  $55 = (11 \times 10)/2$  additional normalizers. Fig. 4 shows a complete set of  $K = 66$  normalizers obtained this way. They are ordered according to their weights obtained by the sum of AdaBoost weights of their corresponding weak classifiers by training the AdaBoost strong classifiers on the TU Darmstadt pedestrian dataset [1] (see Section 8 for more details). A larger weight indicate that a given normalizer makes more contribution in helping NOCD distinguish true positive from clutter background. The weight order of the normalizers confirms the simplicity principle that guided our design of normalizers in that simpler normalizers are usually more significant. However, the weights of the normalizers are also influence by their ability to match well to noise, which may be image class specific. For example, straight lines in horizontal and vertical direction belong to a common background clutter in inner city images as the images of the TU Darmstadt pedestrian dataset.

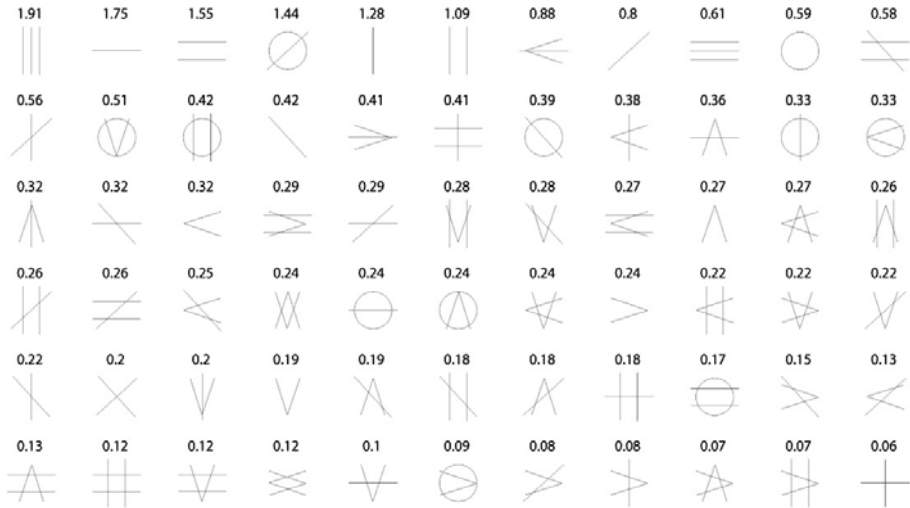


Fig. 4. Our 66 normalizers displayed in order of their weights

For each contour part of a target model  $c_{ij}$ , we resize the normalizers to let them have the same bounding box as the contour part  $c_{ij}$ . Consequently, the resized normalizers cover the same area. Fig. 5 shows the resized normalizers generated for each bundle of the human model.

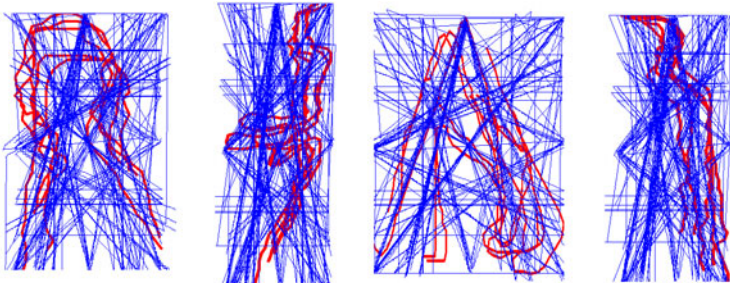


Fig. 5. Human model normalizers. The resized normalizers for four part bundles are shown in blue. The red curves are the original model parts for each bundle.

## 8 Experimental Evaluation of Detection Rate

In this section we compare object detection performance of the proposed normalized oriented chamfer distance (NOCD) to the oriented chamfer distance (OCD) and to chamfer distance on standard test datasets. The detection method is described in Section 6. We use exactly the same flexible models and the same experimental settings for both methods. In particular, for each image, the edge

map was computed by the canny edge detector with the same threshold. The chamfer distance was computed exactly as defined in formula (5). The same constants  $\tau$  and  $\lambda$  were used to truncate the distance transform and linearly combine the distance and orientation terms when calculating the oriented chamfer distance. Results are quantified in terms of detection rate. We use the standard PASCAL criterion to identify correct detections. A detection is regarded as correct if the area of the intersection of the bounding box containing the detected object with the ground truth bounding box is at least 50% of the area of their union.

*TU Darmstadt Pedestrian Dataset.* Human detection is very challenging for shape-based matching methods, because in many poses the shape of human contours is relatively simple. In surveillance images, there is often a complex background, while humans are relatively small, which also increases the chance for an accidental matching.

TU Darmstadt pedestrian dataset [1] consists of several series of video images containing side-view humans. It provides two training datasets, one has 210 images and another has 400 images. In our experiment, we use training 400 dataset for the training of NOCD. After that, we test both NOCD and OCD on the test dataset with 250 images. The 250 test images are significantly more challenging than the 400 training images. To handle the variance of the human shape caused by people walking in opposite directions, we flip our model with respect to vertical axis, and take the best score of the original and flipped models. Consistent with the results of the  $\lambda$  learning procedure reported in Shotton [17], we also observed that detection accuracy of oriented chamfer distance increases when  $\lambda$  becomes larger. In all human detection experiments, we used  $\lambda = 0.8$  for both OCD and NOCD, which was the best performing. As it is often the case in AdaBoost applications, we discarded weak classifiers with very small weights. After training phase, we retained only 37 normalizers with largest weights to form the strong learner for each model contour part. This allows us to reduce the object detection cost complexity.

The detection rate is shown in Table 1. We observe that the proposed NOCD nearly doubled the detection rate of OCD on the 250 test images. The improvement is very significant given the fact that the detection rate of OCD is very low: 35.2%.

Several detection results are displayed in Fig. 1. As they illustrate OCD fails when the human contours are broken and distorted while at the same time the background is cluttered. This is exactly when the proposed NOCD performs extremely well. We also report the performance of pure chamfer distance in Table 1 in order to show that OCD performs significantly better than chamfer distance on this dataset. Further, we include the detection rates of state-of-the-art approaches estimated from graphs reported in [1]. We observe that our detection rate is compatible to a popular appearance based detector, HOG [23]. We stress that our approach is still a matching approach. Andriluka et al. [1] obtained the currently best performance on this dataset. It is obtained by an approach specifically designed for pedestrian detection that utilizes a sophisticated statistical



**Table 1.** Detection rate on Test 250 of the TU Darmstadt Pedestrian Dataset. The proposed NOCD doubled the OCD detection rate with exactly the same contour model.

Chamfer distance	4.4%	HOG [23]	72%
OCD	35.2%	4D-ISM [24]	81%
proposed NOCD	70%	Andriluka et al. [1]	92%

**Table 2.** Detection rate on Cow Dataset

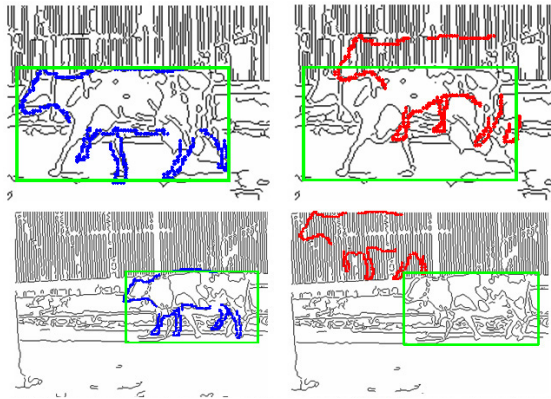
Chamfer distance	73.9%	proposed NOCD	91.0%
OCD	73.9%	Zhu et al. [19]	88.2%

inference framework and learning to handle articulations; both not present in our approach. Similarly, the approach in [24] is designed to handle articulations for pedestrian detection.

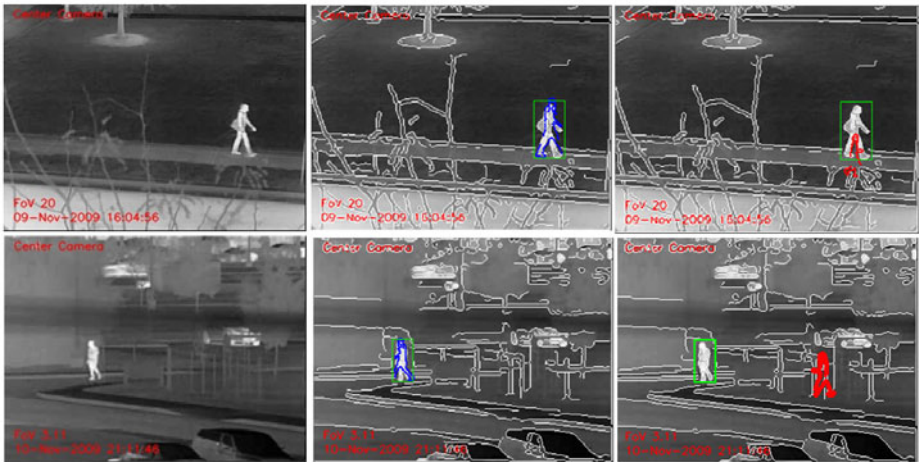
*Cow dataset.* This dataset [5] is from the PASCAL Object Recognition Database Collection. There are 111 images in which cows appear at various positions. Since no training part is provided, we divided the dataset into two parts. We used first 55 images to train our detector, and tested it on the remaining 56 images. Then we trained on the second part, and tested on the first 55 images. This way we are able to report our performance on the whole dataset. The detection rates are shown in Table 2. Again we report a substantial increase in the detection rate by over 17% of NOCD in comparison to OCD. Interestingly, OCD is not able to improve the performance of pure chamfer distance. For this dataset, we used  $\lambda = 0.2$ , which indicates that the orientation information is not particularly useful. This is most likely due to a particular kind of background clutter present in this dataset as can be seen in the example result images in Fig. 6. The areas with dense vertical lines in the edge maps confused oriented chamfer matching. Oriented chamfer matching could not tell the ground truth location from such noise, since most of the false alarms appear in that area. The proposed NOCD was able to learn the difference between such noise and the true targets. For images with little clutter in the background, both OCD and NOCD performed equally well.

The performance of NOCD on this dataset also compares favorably to a very sophisticated learning and inference approach published very recently by Zhu et al. [19]. This comparison may not be quite fair, since this approach uses one-example learning, while our flexible cow model is constructed from 5 cow contours. However, on the other hand our detection algorithm is a simple max-sum. Thus, we do not employ any sophisticated inference in the detection process.

*Infrared images.* Without extra training, we use the same human model and the same normalizers as for TU Darmstadt Pedestrian dataset to carry out several tests on infrared images. In these images, humans are small, about  $60 \times 40$  pixels, which increase the possibility of misalignment to background. Some detection results are shown in Fig. 7.



**Fig. 6.** Example detection results on the cow dataset. Left column NOCD. Right column OCD. Green rectangle denotes the ground truth object location.



**Fig. 7.** Detection result for infrared images. The original images are in the first column. The second column shows result of NOCD while the third column shows the results of OCD. Blue and red dots represent the corresponding parts of the model. Green rectangle denotes the ground truth bounding box. The edge map is overlaid in white on the original images.

## 9 Conclusions

By adding the term of orientation in the evaluation of the score, oriented chamfer distance is more robust to accidental alignment to the background noise than chamfer distance. However, as our experimental results clearly demonstrate this still does not solve the problem of matching to cluttered background, which

often leads to a better score than the score at true object location. The proposed NOCD provides a solution to this problem by utilizing AdaBoost to learn normalization of OCD. The key idea is to compare the chamfer matching score of a given template to scores of a set of normalizers. The obtained ratios are interpreted as weak learners, and the strong learner obtained by AdaBoost is interpreted as a normalized OCD. Based on specific application, the proposed method could be modified by replacing oriented chamfer distance with oriented Hausdorff distance, or using sparse logistic regression instead of Adaboost in training phase.

## Acknowledgments

The work has been supported by the NSF Grants IIS-0812118, BCS-0924164, the AFOSR Grant FA9550-09-1-0207, and the DOE Award 71498-001-09.

## References

1. Andriluka, M., Roth, S., Schiele, B.: People-Tracking-by-Detection and People-Detection-by-Tracking. In: CVPR (2008)
2. Barrow, H.G., Tenenbaum, J.M., Bolles, R.C., Wolf, H.C.: Parametric correspondence and chamfer matching: Two new techniques for image matching. In: Proc. 5th Int. Joint Conf. Artificial Intelligence, pp. 659–663 (1977)
3. Gavrila, D.M., Munder, S.: Multi-cue Pedestrian Detection and Tracking from a Moving Vehicle. *International Journal of Computer Vision* 73(1), 41–49 (2007)
4. Leibe, B., Seemann, E., Schiele, B.: Pedestrian Detection in Crowded Scenes. In: IEEE Conf. on Computer Vision and Pattern Recognition (2005)
5. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: ECCV 2004 Workshop on Statistical Learning in Computer Vision (2004)
6. Stenger, B., Thayananthan, A., Torr, P.H.S., Cipolla, R.: Model-based hand tracking using a hierarchical bayesian filter. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 28(9), 1372–1384 (2006)
7. 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. 3952, pp. 575–588. Springer, Heidelberg (2006)
8. Opelt, A., Pinz, A., Zisserman, A.: Incremental learning of object detectors using a visual shape alphabet. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 1, pp. 3–10 (2006)
9. Heitz, G., Elidan, G., Packer, B., Koller, D.: Shape-Based Object Localization for Descriptive Classification. *International Journal of Computer Vision* 84(1) (August 2009)
10. Van Herk, M.: Image Registration Using Chamfer Matching. In: Handbook of medical imaging: processing and analysis. Academic Press, London (2000)
11. Nomira, O., Abdel-Mottaleb, M.: Hierarchical contour matching for dental X-ray radiographs. *Pattern Recognition* 41(1), 130–138 (2008)
12. Faisan, S., Passat, N., Noblet, V., Chabrier, R., Meyer, C.: Topology Preserving Warping of Binary Images: Application to Atlas-Based Skull Segmentation. In: Metaxas, D., Axel, L., Fichtinger, G., Székely, G. (eds.) MICCAI 2008, Part I. LNCS, vol. 5241, pp. 211–218. Springer, Heidelberg (2008)

13. Wu, B., Nevatia, R.: Detection and Segmentation of Multiple, Partially Occluded Objects by Grouping, Merging, Assigning Part Detection Responses. *International Journal of Computer Vision* 82(2), 185 (2009)
14. Thayananthan, A., Stenger, B., Torr, P., Cipolla, R.: Shape context and chamfer matching in cluttered scenes. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 127–133 (2003)
15. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 24(24), 509–522 (2002)
16. Borgefors, G.: Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Trans. Pattern Analysis and Machine Intell.* 10(6), 849–865 (1988)
17. Shotton, J., Blake, A., Cipolla, R.: Multi-scale categorical object recognition using contour fragments. *IEEE Trans. Pattern Analysis and Machine Intell.* 30(7), 1270–1281 (2008)
18. Freund, Y., Schapire, R.: A decision theoretic generalisation of online learning. *Computer and System Sciences* 55(1), 119–139 (1997)
19. Zhu, L., Chen, Y., Yuille, A.: Learning a Hierarchical Deformable Template for Rapid Deformable Object Parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 99(1) (2009)
20. Lee, Y., Grauman, K.: Shape Discovery from Unlabeled Image Collections. In: *IEEE Conf. on Computer Vision and Pattern Recognition*, Miami, FL, USA, pp. 2254–2261 (June 2009)
21. Torralba, A., Murphy, K., Freeman, W.: Sharing Features: Efficient Boosting Procedures for Multiclass Object Detection. In: *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 762–769 (2004)
22. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: *IEEE Conf. on Computer Vision and Pattern Recognition* (2001)
23. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *IEEE Conf. on Computer Vision and Pattern Recognition* (2005)
24. Seemann, E., Schiele, B.: Cross-articulation learning for robust detection of pedestrians. In: Franke, K., Müller, K.-R., Nickolay, B., Schäfer, R. (eds.) *DAGM 2006*. LNCS, vol. 4174, pp. 242–252. Springer, Heidelberg (2006)
25. Trinh, N.H., Kimia, B.B.: Category-Specific Object Recognition and Segmentation Using a Skeletal Shape Model. In: *Proceedings of the Twentieth British Machine Vision Conference*, London, UK (September 2009)
26. Huttenlocher, D.P., Klanderma, G.A., Rucklidge, W.J.: Comparing Images Using the Hausdorff Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(9), 850–863 (1993)
27. Olson, C.F., Huttenlocher, D.P.: Automatic Target Recognition by Matching Oriented Edge Pixels. *IEEE Transactions on Image Processing* 6(1), 103–113 (1997)

# Geometry Construction from Caustic Images

Manuel Finckh, Holger Dammertz, and Hendrik P.A. Lensch

Institute of Media Informatics, Ulm University  
89081 Ulm, Germany

**Abstract.** In this work we investigate an inverse geometry problem. Given a light source, a diffuse plane and a caustic image, how must a geometric object look like (transmissive or reflective) in order to project the desired caustic onto the diffuse plane when lit by the light source? In order to construct the geometry we apply an analysis-by-synthesis approach, exploiting the GPU to accelerate caustic rendering based on the current geometry estimate. The optimization is driven by simultaneous perturbation stochastic approximation (SPSA). We confirm that this algorithm converges to the global minimum with high probability even in this ill-posed setting. We demonstrate results for precise geometry reconstruction given a caustic image and for reflector design producing an intended light distribution.

## 1 Introduction

The automatic construction of geometric objects from a predetermined property is an important engineering task. We address the problem of constructing a transmissive or reflective surface, that, given a predefined light position, creates an a priori defined caustic image. This task of creating a specific caustic occurs in the design of headlights, of parabolic concentrators for solar cells or interior design, see for example Figure 1. The same approach can also be used to reconstruct the surface geometry of a real object given only an image of its caustic. Even though this is an ill-posed problem we show how in many cases reasonable reconstructions can be achieved.

The geometry estimation follows an analysis-by-synthesis approach. The procedure starts with an initial surface whose geometry is subsequently optimized to minimize the *mean squared error* (MSE) between the target caustic and the current caustic image. The MSE is the only measure applied to determine the quality of the constructed geometry. A standard optimization algorithm for such a problem would be the *simulated annealing* (SAN) algorithm [1, 2]. However, our results show that the *simultaneous perturbation stochastic approximation* (SPSA) optimization algorithm [3] is more robust and converges much faster in this setting.

One contribution of our work is the use of SPSA as a *global* optimizer for this kind of problem. For the evaluation of the objective function we present a specialized and optimized implementation that exploits GPUs for a fast evaluation. It efficiently renders single bounce reflections or refractions. The proposed



**Fig. 1.** The left image shows four of our optimization targets for which we generated geometry with our proposed optimization approach. The second and third image show a physically correct light transport simulation illustrating the results. For the living room scene we optimized a glass table to cast the predefined caustic. The letters E C C V in the right image are caustics from parabolic reflectors with a small embedded light source. The last image shows one half of such an optimized parabolic reflector.

optimization framework is flexible to work on connected or disconnected triangle meshes and further can operate on  $C^2$  continuous B-spline surfaces. One unique feature of our method compared to other methods like [4, 5] is the degree of freedom we can deal with due to utilizing SPSA for optimization. In the case of the parabolic reflectors in Figure 1 the B-spline patches have about 4000 control points which have to be optimized. We explore the use of our framework for reflector design, for geometry reconstruction of water surfaces and for light concentrating glass objects.

## 2 Related Work

A recent survey for various ways of reconstructing specular and transparent geometry has been assembled by Ihrke et al. [6]. Direct geometry measurement techniques are based on structured illumination [7, 8] and multiple input images. They apply shape from distortion [9–11], shape from specular highlights [12, 13], optical tomography [14], or inverse ray-tracing [15, 16]. Our reconstruction method falls into the last category but uses a single intensity distribution image as input and therefore requires optimization.

Morris et al. [17] demonstrated the reconstruction of a water surface by utilizing two cameras and a known pattern placed under the water surface. With some restrictions to the setup, for instance secondary refractions or reflections have to be suppressed, they were able to reconstruct the surface correctly. Reconstructing a water surface is not the scope of this work, nevertheless our methods could be applied to this task.

Kutulakos et al. [18] investigate the theoretical background of reconstructing arbitrarily-shaped specular scenes. They reduced the problem of 3D shape reconstruction to reconstruction of light paths that cross the image plane. They showed, that it is impossible to reconstruct a light path when the light is reflected or refracted more than twice. In all other cases, three viewpoints are enough for

successful reconstruction. This insight limits the generality of our approach, as a single caustic image allows us to consider a single surface interaction only. In addition Ramamoorthi et al. [19] provides a theoretical framework for inverse rendering problems.

Patow [20] and Patow et al. [21, 22] investigated the specific problem of reflector design to obtain surfaces that produce an intended light distribution for a given light source position. They applied an analysis-by-synthesis approach using a brute force search and SAN in order to optimize the reflector for the distribution. Early work on the reflector design problem was done by Neubauer [23], Caffarelli et al. [24], and Wang [25, 26].

Recent approaches directly operating on NURBS-surfaces and utilizing an analysis-by-synthesis approach are presented by Anson et al. [4] and Mas et al. [5]. However, the search space was restricted to only two dimensions and rotationally symmetric reflectors, and four dimensions respectively.

Recently, Weyrich et al. [27] presented a method of fabricating micro geometry with custom reflectance probability. They first computed a set of micro facets which produces the selected reflectance distribution. However, the resulting micro facets are not connected, so they utilized a SAN optimization process to arrange them in a (nearly) tileable way. Our approach generates comparable results on disconnected meshes but is flexible enough to optimize watertight surfaces as well. Beyond that, our SPSA-based system works efficiently for both reflective and refractive surfaces.

### 3 Optimization Framework

Caustics occur when a specular or refractive object focuses light onto a diffuse surface. Caustics are often caused by water surfaces, glass objects, such as lenses, or concave mirrors as one finds them in headlights. In photo-realistic image synthesis a common task is to simulate these caustics [28]. In this work, we ask for the inverse problem: Given the caustic image, what is the geometry of the caustic generating surface? During optimization we apply a simplified rendering system (Section 5) which ignores multiple scattering inside the reflector or refractor.

We can not use methods as in [17, 18], because the directional information of the incident light onto the diffuse surface is not available in a single caustic image. Hence, we are not able to directly reconstruct the light paths which would allow us to reconstruct a single reflective or refractive surface directly. Furthermore, as the incident direction to the diffusor is unknown the problem is not well-defined, even the simple case where all the light is focused on a single point is not uniquely solvable. The solution could be a small lens directly in front of the light source, or a huge lens further away from the light source.

Therefore, we use an analysis-by-synthesis approach to find an appropriate solution. The reflecting or transmitting surface is represented as a triangle mesh or in most examples as  $C^2$  continuous B-spline patch which is either initialized as a planar surface, or a parabolic surface (mainly used for reflector optimization).

The number of control points is arbitrary, for the results presented here we use  $12^2$  up to  $64^2$  control points. The control points are fixed in their  $xy$ -position, only the  $z$ -coordinate is modified during optimization. Allowing for varying  $xy$ -coordinates would not only increase the dimensionality of the problem, it would also complicate the optimization process, i.e., preventing self occlusion would not be trivial in such a setting. The B-spline patch is described by one state vector  $\theta$ , with dimensionality  $p$  equal to the number of control points. Hence, we search for an optimal solution vector  $\theta^*$  in the problem space  $\Theta \subseteq \mathbb{R}^p$ . The optimal solution is defined by the global minimum of the objective function  $L(\cdot)$ , which in our case is the MSE between the current caustic image and the target caustic image.

The missing pieces to fully describe our framework are the employed optimization algorithm and the evaluation of the objective function which includes the costly computation of the caustic. These points are discussed in the next two sections.

### 4 Optimization Using SPSA

The optimization is carried out by performing a random walk on the problem space  $\Theta$ . In each iteration a new candidate solution  $\theta_{k+1}$  is computed by adding a specific step vector to the current state vector  $\theta_k$ . In contrast to simulated annealing which takes a completely random approach, the SPSA algorithm computes an approximate gradient to determine the best search direction in each iteration.

The SPSA algorithm belongs to the family of *stochastic approximation* (SA) algorithms [29]. The basic form of the SA algorithm when there is no analytic gradient available is the Kiefer-Wolfowitz finite-difference SA (FDSA) algorithm [30, 31]. The disadvantage of this algorithm is that it needs  $2p$  objective function evaluations in order to approximate a gradient. Introduced by Spall [3], the SPSA algorithm overcomes this disadvantage. It consumes only two objective function evaluations in each iteration in order to approximate a gradient regardless of the dimensionality of the problem.

**Gradient Approximation.** The idea is to randomly perturb all elements of  $\theta_k$  to obtain two (probably noisy) measurements of the objective function. More formally, let  $y(\cdot)$  denote a noisy measurement of  $L(\cdot)$ , i.e.,  $y(\cdot) = L(\cdot) + noise$ . Each component  $i$  of the  $k$ -th approximate gradient  $g_k(\theta_k)$  is now determined by,

$$g_{ki}(\theta_k) = \frac{y(\theta_k + c_k \Delta_k) - y(\theta_k - c_k \Delta_k)}{2c_k \Delta_{ki}} . \tag{1}$$

A simple choice for the  $\Delta_k$  random vector is to use a Bernoulli $\pm 1$  distribution for each component  $\Delta_{ki}$  of the vector (with probability 1/2 for each  $\pm 1$ ), in general each  $\Delta_{ki}$  has to be independent and symmetrically distributed about 0 with finite inverse moments  $E(|\Delta_{ki}|^{-1}) < \infty$  for all  $k, i$  [32]. The so-called gain



**Table 1.** Choice of the gain sequences

$a_k$	$a/(A+k+1)^\alpha$
$\alpha$	0.602 (practically effective), 1.0 (asymptotically optimal)
$A$	< 10% of maximum (expected) iterations
$a$	$a/(A+1)^\alpha \cdot g_0(\theta) \approx$ smallest desired change among elements in $\theta$
$c_k$	$c/(k+1)^\gamma$
$\gamma$	0.101 (practically effective), 1/6 (asymptotically optimal)
$c$	small positive number $\approx$ standard deviation of the measurement
$\Delta_k$	Bernoulli $\pm 1$ distribution

sequence  $c_k$  controls the distance between the sample points. It is monotonically decreasing in each iteration to ensure high quality gradients when approaching the optimum.

The optimization is carried out by moving along this approximated gradient  $g_k$ , formally,

$$\theta_{k+1} = \theta_k - a_k g_k(\theta_k), \quad (2)$$

where the  $a_k$  is another gain sequence generating monotonically decreasing step lengths.

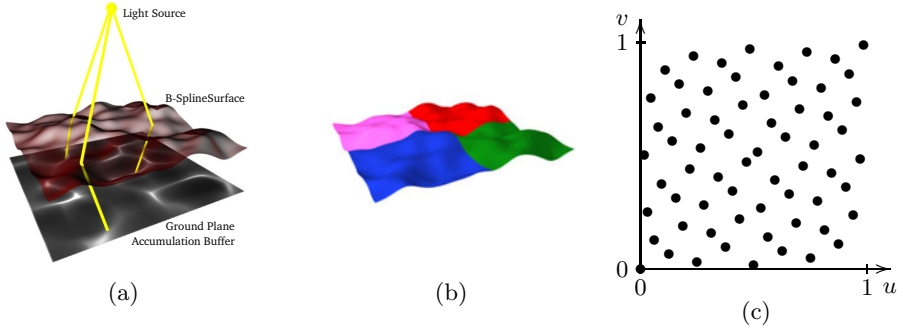
**The Gain Sequences.** Unfortunately, there is no generally optimal choice for the gain sequences  $a_k$  and  $c_k$  and the random vector  $\Delta_k$  in practice, only theoretically optimal choices are at hand. In [33], some suggestions are given how to tune these parameters in order to improve convergence, for a summary see Table 1. The specific choice of SPSA parameters for our purpose is discussed in Section 6.

**Convergence.** In general, the order of the error is  $\epsilon = \mathcal{O}(k^{-1/3})$  [3, 34, 35]. This is only the local convergence rate of the algorithm. But as pointed out in [36] SPSA may work as global optimizer without adding an extra random vector to the SA-recursion (Eq. 2) [37].

## 5 Caustic Rendering on the GPU

As the objective function has to be evaluated several thousand or even million times, a fast implementation of the caustic rendering step is crucial. The main part of the evaluation of the objective function is the synthesis of the caustic (about 99% of the computation time is spent there).

The caustic is synthesized by Monte Carlo light transport simulation [28] ignoring multiple interactions. We can optimize this simulation specifically for our task. The simple scene geometry allows for omitting self occlusion, and further allows direct handling of the ground plane as accumulation buffer. For efficient simulation we split the B-spline into multiple cubic Beziér patches [38] (see Figure 2b). The splitting computation is done on the CPU, the resulting Beziér



**Fig. 2.** Illustration of the rendering process. In (a) the basic process is illustrated. Starting from a light source, light rays are connected to the B-spline surface, refracted (or reflected) and intersected with the ground plane accumulation buffer. (b) shows how, for sampling purposes, the B-spline surface is split into multiple Beziér patches and (c) shows 64 Hammersley points used to generate sample points on each Beziér patch.

patches are then transferred to GPU memory and the remaining computation of the caustic is done on the GPU. The basic steps performed on the GPU are the following (illustrated in Figure 2a):

We start a fixed number of threads (i.e. 64) on the GPU. The number of samples we take for each Beziér patch is also fixed and a multiple of the number of threads (so each thread has to take  $\#work = \#samples / \#threads$  samples). For the generation of the sample points we use the Hammersley quasi-Monte Carlo point set [39] (see Figure 2c). They are well distributed and very simple to compute,

$$(x, y) = (i/n, (reverse\ bits)(i)/0x100000000LL), \tag{3}$$

where  $n = \#samples$ , and  $i = sample\_index$ .

Each thread computes now:

- I. initialize ground plane accumulation buffer with zeros
- II. for each Bezier patch:
  1.  $sample\_index = k + thread\_num * work$ , where  $k$  in range  $(0, work)$
  2. compute point  $(x, y)$
  3. transform into 3D point on Bezier patch
  4. connect point to light source and compute refraction ray
  5. intersect with ground plane
  6. accumulate contribution with correct weighting

Essential for the correctness of the simulation is the weighting of the samples. As we directly sample points on the surface patches we have to weight them according to the projected differential surface area and the squared distance  $r$ .

The differential surface area is given by the length of the cross product of both directional derivatives (length of the surface normal  $n$ ), and the projection results in an additional cosine factor (between normalized surface normal  $\hat{n}$  and direction  $\hat{d}$  to the light source),

$$w = \frac{1}{r^2} \|n\| \cdot |\langle \hat{n} \cdot \hat{d} \rangle|. \quad (4)$$

**Additional Optimization.** The accumulation of the contribution in step 6 results in random memory read-write access. Further the different threads may write to the same part of the accumulation buffer, which either leads to the need of atomic writes or a falsified computation of the caustic. So we simply save the contribution along with the pixel index in a consecutive array. The array is transferred to the main memory and the final caustic image is then assembled by the CPU. By asynchronously calling the CUDA kernel this computation along with the additional memory transfer can be done mostly synchronous to the GPU computation.

**Efficiency of the method.** For example the water surface (see Section 6) the B-Spline is split into 81 Beziér patches, each patch is sampled with 2048 samples which results in 165888 samples per objective function evaluation. 1000 objective functions are evaluated in less than 4 seconds, which results in 30 minutes overall run time for convergence (System: Intel Core 2 Duo E6850 with 3 GHz and NVIDIA GeForce GTX 285).

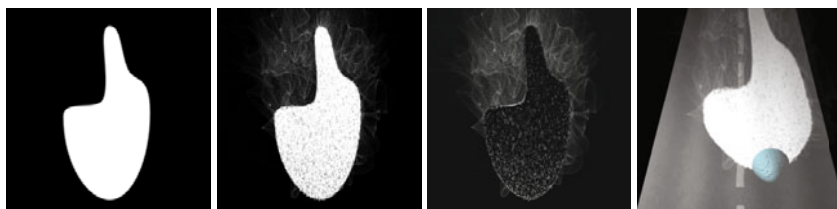
## 6 Results

We apply our framework to a set of different scenarios.

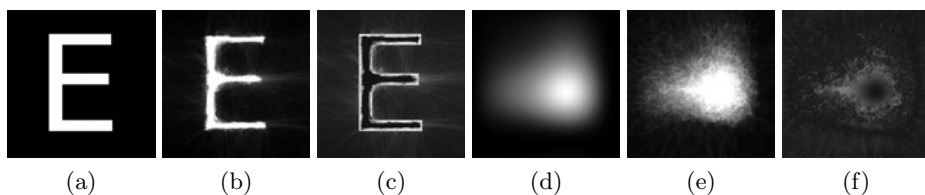
**Designing Reflective and Refractive Concentrators.** In Figure 1, we design refractive and reflective surfaces to generate sharp, high contrast patterns at a specific focus plane. The optimization clearly renders the letters for the glass table and the parabolic reflector, but can only approximate the sharp transition. Due to the  $C^2$  continuity, the optimized caustic cannot perfectly match the original which leads to some background noise in those areas which are intended to be black. The height variation necessary to produce the output is surprisingly small compared to the object size.

**Headlight Design.** A real-world application is shown in Figure 3. Here, a parabolic head light is optimized to cast a non-blinding, almost homogeneous spot of predefined shape onto the street. Note the even intensity distribution in the illuminated region which avoids the hot spot close to the light source typically generated by headlights. Additional results can be seen in Figure 4.

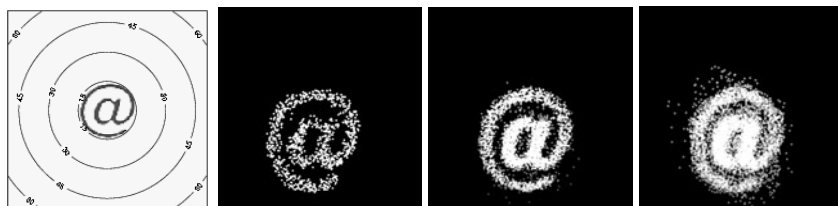
**Optimizing Reflectance Distributions.** Inspired by Weyrich et al. [27], we further investigate generating a specific radiance distribution of an almost planar, reflective surface for directional illumination. We demonstrate the flexibility of



**Fig. 3.** Headlamp Design – from left to right: Predefined shape of the light cone when projected onto the street. Our resulting distribution after optimization. The difference to the original distribution. Illustration of the result.

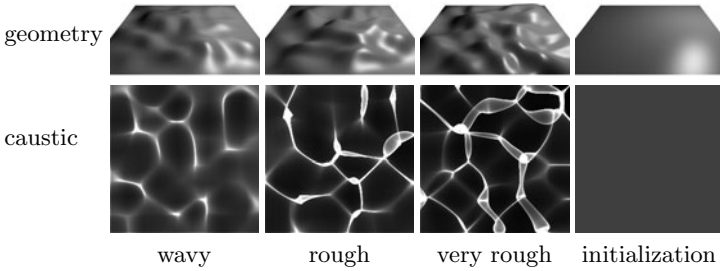


**Fig. 4.** Results for parabolic reflector design, (a,d) show the target distribution, (b,d) the distribution of the resulting reflector, and (c,f) the difference to the target distribution.



**Fig. 5.** Results for the micro facet optimization. The first image shows the target distribution, the second image the result by Weyrich et al. [27] with single disconnected quads. The third image shows the result of applying our optimization approach to disconnected quads and the last image on a connected triangle mesh. Note that Weyrich et al. directly samples normals from the target distribution and generates corresponding quads, hence there is no sample outside the target distribution. With our method it is possible to directly optimize a closed triangle mesh. We improved our results by adding a minimal Gauss-blur to the target distribution as it introduces a well-defined gradient to the error function.

our framework and compare the performance for connected and unconnected triangle meshes. As shown in Figure 5, optimizing unconnected quads yields results slightly inferior to Weyrich et al., because in this case the normals can be directly sampled from the target distribution and need not to be optimized. However, with our approach it is possible to optimize a closed triangle mesh which is not possible with their method.



**Fig. 6. Top:** Visualization of the ground truth geometry for the global convergence test. **Bottom:** Resulting caustics from the above geometry that were fed into the optimization process. For these tests the optimization is always initialized with a flat surface, which results in an average grey caustic.

**Geometry Reconstruction for Water Surfaces.** In Figure 6, we show the caustics generated by water surfaces of varying roughness. Our system generates geometry that reproduces the caustics up to a small error. The Figures 8, 9, and 10 demonstrate the relationship between the MSE of the caustic images and the error of the geometry. The geometry error was computed by sampling the surface at 10000 fixed locations and summing up the squared differences to the original surface. The size of the surface is 10 by 10cm. The results show that the optimized geometry most often corresponds well to the original surface.

## 7 Discussion

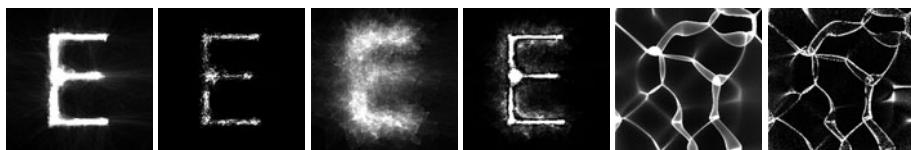
To generate the results we used SPSA with the standard gain sequences  $a_k$  and  $c_k$  as given in Table 1 with  $\alpha = 0.602$  and  $\gamma = 0.101$ . The asymptotically optimal values of  $\alpha = 1.0$  and  $\gamma = 1/6$  would lead to faster local convergence, however, at the cost of greatly reducing the probability of convergence to a global optimum. The parameters  $a$  and  $c$  are adapted to each specific problem and are experimentally chosen observing the beginning of the optimization. The choice of  $c$  influences the gradient approximation and depends on the noise of the objective function (Eq. 1). One has to trade off a noisy vs. artificially smoothed gradient estimation. The value of  $a$  controls the step size and thus the convergence behavior. Often, if the initial curve of the MSE is too smooth, as it should be at the end of an optimization, the algorithm directly converges to a nearby local minimum.  $a$  needs to be increased to allow for a more random exploration of the error landscape. Choosing  $a$  too large though will slow down convergence.

**Global Convergence.** The SPSA algorithm only guarantees a probabilistic convergence to a global optimum [36]. For decent choices of parameters  $a$  and  $c$ , most often, the global optimum in the water surface test cases was found with the first sequence (see Figure 6, 8, 9, and 10). Whenever the final MSE was still too large, as in the failure case in Figure 10, we restarted the optimization with a different random seed, which was never required more than twice in all

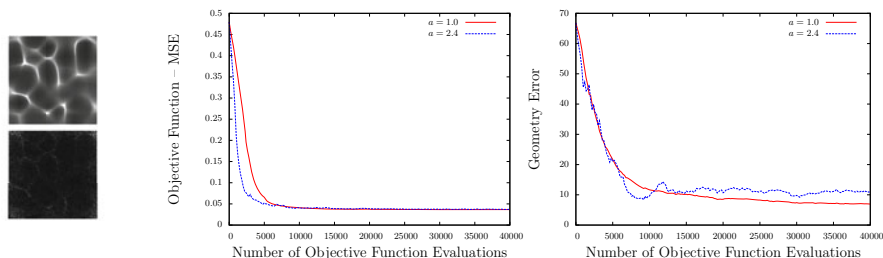
presented cases and could be automatized. Our experimental results verify that SPSA is suitable for global optimization in our setting.

**SPSA vs. Simulated Annealing.** The good convergence probability of SPSA is in contrast to SAN which rarely converged to a decent optimum in bearable time. The drawback of SAN is that it does not explore the error landscape in a controlled way. It randomly samples the neighboring space and can not walk into a specific direction as it does not exploit any gradient information, not even an approximation. The best results with SAN were achieved for the smooth test case of Figure 6 where the initial solutions is already near the global optimum. With SAN, we obtained decent results on the unconnected triangle mesh in Figure 5, but it completely failed at optimizing the parabolic reflectors, probably due to the complexity of the problem.

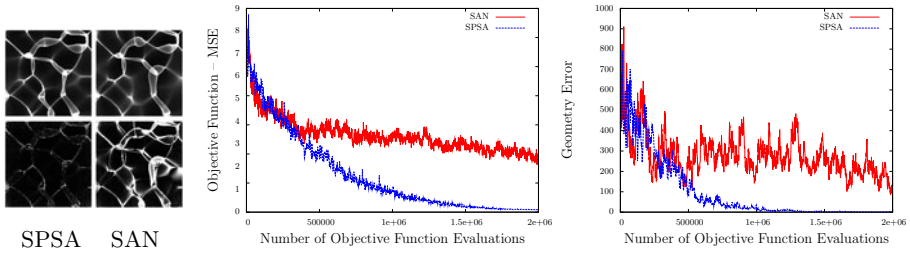
**Manufacturing.** We did not manufacture the reflectors and refractors we optimized, but we verified our results by the means of physical light transport simulation (see Figure 1). We also simulated the precision of a typical milling machine and randomly perturbed the estimated geometry (see Figure 7).



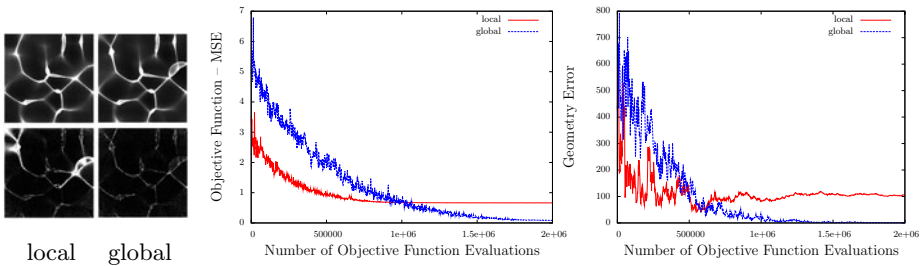
**Fig. 7.** Results for the precision simulation, from left to right: E-reflector with randomly perturbed control points ( $\pm 0.02\text{mm}$ ) and the difference to the result with no induced error. Again the E-reflector now with a random perturbation of  $\pm 0.2\text{mm}$ , the "E" nearly vanishes in the noise. The last two images show the SPSA optimization of the very rough test case after 250000 iterations, the geometry error corresponds to an average error of about 0.1mm.



**Fig. 8.** On this input data the  $a$  parameter can be chosen arbitrarily out of a large interval. The reason for this is, that the next nearby local optimum is identical to the global optimum. However, with even smaller values for  $a$  the convergence rate of the algorithm would be greatly reduced and with larger ones the algorithm might jump over the global optimum. Note that the remaining difference in the geometric error is extremely small and corresponds to an average error of less than 0.01mm.



**Fig. 9.** SPSA and SAN applied to the very rough test case given in Figure 6. SPSA converges to the correct result, SAN converges much slower. The left images show the result after the optimization run (top) and the difference to the target caustic (bottom). We can also see by comparing the objective function graph with the associated geometry error graph that a smaller MSE of the objective function does not always result in a smaller geometry error. This indicates local optima. Also note the smooth MSE graph at the end of the optimization process, indicating local (and in this case global) convergence.



**Fig. 10.** Illustration of a failure case, where SPSA did not find the global optimum at the first but the second run. In the left columns the top images show the optimization result and bottom images the difference to the original caustic (compare to Figure 6). Note the local minimum of the geometry error of the failure case at about 500000 and the monotonic decrease of the corresponding objective function. The algorithm is clearly converging to a local minimum.

## 8 Conclusion

The proposed system successfully produces surface geometry that matches a specific reflection or refraction distribution pattern. It can be used to reconstruct geometry from a given caustic image or to shape special purpose reflectors. Key aspects of our efficient optimization are the use of the SPSA algorithm which ensures global convergence with high probability as well as the GPU accelerated caustic rendering approach. Our framework can cope with various surface representations, allowing for the flexibility of unconnected triangles or for the slightly more restrictive but easy to manufacture continuous B-spline surfaces.

Currently, our framework is restricted to single reflective or refractive surfaces. Adding solid objects with multiple scattering events results in a much more complex evaluation of the objective function. It will be a challenge to deal with the additional ambiguities which result in much more distinct local optima of the objective function [18]. However, the optimization framework based on SPSA is independent of the rendering technique and can be exchanged with other methods.

## Acknowledgments

This work has been partially funded by the DFG Emmy Noether fellowship (Le 1341/1-1) and by an NVIDIA Professor Partnership Award.

## References

1. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computer machines. *Journal of Chemical Physics* 21, 1087–1091 (1953)
2. Kirkpatrick Jr., S., Gellatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* 220(4598), 671–680 (1983)
3. Spall, J.C.: A Stochastic Approximation Algorithm for Large-Dimensional System in the Kiefer-Wolfowitz Setting. In: *IEEE Conf. Decision Contr.*, pp. 1544–1548 (1988)
4. Anson, O., Seron, F.J., Gutierrez, D.: NURBS-based inverse reflector design. In: *Proceedings of CEIG 2008*, pp. 65–74 (2008)
5. Mas, A., Martín, I., Patow, G.: Fast inverse reflector design (FIRD). *Comput. Graph. Forum* 28, 2046–2056 (2009)
6. Ihrke, I., Kutulakos, K.N., Lensch, H.P.A., Magnor, M., Heidrich, W.: State of the art in transparent and specular object reconstruction. In: *STAR Proceedings of Eurographics*, pp. 87–108 (2008)
7. Morris, N.J.W., Kutulakos, K.N.: Reconstructing the surface of inhomogeneous transparent scenes by scatter-trace photography. In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)* (2007)
8. Hullin, M.B., Fuchs, M., Ihrke, I., Seidel, H.P., Lensch, H.P.A.: Fluorescent Immersion Range Scanning. *ACM Trans. on Graphics (SIGGRAPH 2008)* 27, 87:1–87:10 (2008)
9. Oren, M., Nayar, S.K.: A theory of specular surface geometry. *International Journal of Computer Vision (IJCV)* 24, 105–124 (1996)
10. Tarini, M., Lensch, H.P.A., Goesele, M., Seidel, H.P.: 3d acquisition of mirroring objects. *Graphical Models* 67, 233–259 (2005)
11. Efros, A.A., Isler, V., Shi, J., Visontai, M.: Seeing through water. In: *Advances in Neural Information Processing Systems* 17, pp. 393–400. MIT Press, Cambridge (2004)
12. Zisserman, A., Giblin, P., Blake, A.: The information available to a moving observer from specularities. *Image and Vision Computing* 7, 38–42 (1989)
13. Wang, J., Dana, K.J.: Relief texture from specularities. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 28, 446–457 (2006)



14. Trifonov, B., Bradley, D., Heidrich, W.: Tomographic reconstruction of transparent objects. In: Proceedings of Eurographics Symposium on Rendering 2006, pp. 51–60 (2006)
15. Ihrke, I., Goldluecke, B., Magnor, M.: Reconstructing the geometry of flowing water. In: Proceedings of IEEE International Conference on Computer Vision (ICCV), pp. 1055–1060 (2005)
16. Miyazaki, D., Ikeuchi, K.: Inverse polarization raytracing: Estimating surface shape of transparent objects. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 910–917 (2005)
17. Morris, N.J.W., Kutulakos, K.N.: Dynamic refraction stereo. In: Tenth IEEE International Conference on Computer Vision (ICCV), pp. 1573–1580 (2005)
18. Kutulakos, K.N., Steger, E.: A theory of refractive and specular 3d shape by light-path triangulation. In: Tenth IEEE International Conference on Computer Vision (ICCV), pp. 1448–1455 (2005)
19. Ramamoorthi, R., Hanrahan, P.: A signal-processing framework for inverse rendering. In: SIGGRAPH 2001: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pp. 117–128. ACM, New York (2001)
20. Patow, G.: Reflector Shape Design From Radiance Distributions. CAD for luminaries. PhD thesis, Universitat Politècnica de Catalunya (2005)
21. Patow, G., Pueyo, X., Vinacua, A.: User-guided inverse reflector design. *Computers & Graphics* 31, 501–515 (2007)
22. Patow, G., Pueyo, X.: A survey of inverse surface design from light transport behavior specification. *Computer Graphics Forum* 24, 773–789 (2005)
23. Neubauer, A.: The iterative solution of a nonlinear inverse problem from industry: design of reflectors. In: Proceedings of the international conference on Curves and surfaces in geometric design, pp. 335–342. A. K. Peters, Ltd., Natick (1994)
24. Caffarelli, L.A., Kochengin, S.A., Oliker, V.I.: On the numerical solution of the problem of reflector design with given far-field scattering data. *Contemporary Mathematics* 226, 13–32 (1999)
25. Wang, X.J.: On the design of a reflector antenna. *Inverse Problems* 12, 351–375 (1996)
26. Wang, X.J.: On the design of a reflector antenna II. *Calc. Var. PDE* 20, 329–341 (2004)
27. Weyrich, T., Peers, P., Matusik, W., Rusinkiewicz, S.: Fabricating microgeometry for custom surface reflectance. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 28 (2009)
28. Veach, E.: Robust Monte Carlo Methods for Light Transport Simulation. PhD thesis, Stanford University (1997)
29. Robbins, H., Monro, S.: A stochastic approximation method. *Annals of Mathematical Statistics* 29, 400–407 (1951)
30. Kiefer, J., Wolfowitz, J.: Stochastic estimation of a regression function. *Annals of Mathematical Statistics* 23, 462–466 (1952)
31. Blum, J.R.: Multidimensional stochastic approximation methods. *Annals of Mathematical Statistics* 25, 737–744 (1954)
32. Spall, J.C.: An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins Apl Tech. Digest*. 19(4), 482–492 (1998)
33. Spall, J.C.: Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Trans. Aerosp. Electron. Syst.* 34(3), 817–823 (1998)
34. Spall, J.C.: Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans. Autom. Control*. 37, 332–341 (1992)

35. Yin, G.: Rates of convergence for a class of global stochastic optimization algorithms. *SIAM J. on Optimization* 10, 99–120 (1999)
36. Maryak, J.L., Chin, D.C.: Global random optimization by simultaneous perturbation stochastic approximation. In: *WSC 2001: Proceedings of the 33rd conference on Winter simulation*, pp. 307–312. IEEE Computer Society, Washington (2001)
37. Jones, M.H.: Jr., White, K.P.: Stochastic approximation with simulated annealing as an approach to global discrete-event simulation optimization. In: *Winter Simulation Conference*, pp. 500–507 (2004)
38. Piegl, L., Tiller, W.: *The NURBS book*, 2nd edn. Springer, New York (1997)
39. Lemieux, C.: *Monte Carlo and Quasi-Monte Carlo Sampling*. Springer, Heidelberg (2009)

# Archive Film Restoration Based on Spatiotemporal Random Walks

Xiaosong Wang and Majid Mirmehdi

Computer Vision Group, Department of Computer Science,  
University of Bristol, Bristol BS8 1UB, UK  
{wang,majid}@cs.bris.ac.uk

**Abstract.** We propose a novel restoration method for defects and missing regions in video sequences, particularly in application to archive film restoration. Our statistical framework is based on random walks to examine the spatiotemporal path of a degraded pixel, and uses texture features in addition to intensity and motion information traditionally used in previous restoration works. The degraded pixels within a frame are restored in a multiscale framework by updating their features (intensity, motion and texture) at each level with reference to the attributes of normal pixels and other defective pixels in the previous scale as long as they fall within the defective pixel's random walk-based spatiotemporal neighbourhood. The proposed algorithm is compared against two state-of-the-art methods to demonstrate improved accuracy in restoring synthetic and real degraded image sequences.

**Keywords:** Video Restoration; Random Walks; Multiscale Refinement.

## 1 Introduction

Archived films suffer damage and quality degradation often through inappropriate storage and wear and tear, but sometimes even at the time of production. The most common types of defects are blotches and scratches which usually appear in one or more (consecutive) frames as black, or white, or semi-transparent regions. However, degrees of degradation and their shape and size can vary due to their random appearance. Quality control and restoration is therefore necessary before such films are broadcastable again and indeed the preferred route to preservation and rebroadcasting is digitisation and automated restoration - a more economical and reversible process compared to the manual and tiresome course of restoration by chemical and physical means, considering the enormous amount of archives there exists. Figure 1 shows an example of a restored frame from a clip we call *Cliff*.

An automated restoration system is usually composed of two modules, defect detection and defect removal. Defect detectors such as [19,15,22,24] provide not only quantitative measures as evidence for quality control but also defect maps which can be used by others to perform defect removal. Thus in this paper, our focus is on restoration and defect removal using defect maps generated from any defect detection work, such as [24].



**Fig. 1.** A degraded frame before and after restoration by the proposed method

One approach to the recovery of a degraded pixel is to replace it with an original corresponding pixel along its projected motion trajectory from (temporal) neighbouring frames. This clearly involves an accurate estimation of the degraded pixel's motion through space and time and helps enforce a local consistency by imposing features besides just image intensities, i.e. motion vectors (leading to consistent optical flows). The chances of more accurate recovery can be increased by recruiting more significant features, e.g. texture features such as the Local Binary Pattern (LBP) [20] (leading to consistent region representation). Unlike previous methods such as [19,23,16,10], we consider multiple features *in an integrated fashion* and show that this provides better restoration than treating the features separately. The computational expense incurred due to the use of more features is an affordable tariff in our archive restoration application where accuracy is of paramount importance.

In order to locate the optimal replacement for a degraded pixel, we establish a region of candidate pixels formed by a number of 3D random walks on the *spatiotemporal* domain, starting from the defective pixel. In [9], spatial-only random walks were applied for noise reduction by taking a weighted average over all spatial pixels visited by the random walks, whereas we select the optimal pixel-exemplar as the pixel which has the maximum likelihood of being the original pixel - as defined by its intensity, motion and texture characteristics - from this dynamically generated spatiotemporal region. We perform this search-and-replace procedure for each degraded pixel in the defect map in a multiscale framework to refine the restored pixels from coarse to fine. This multiscale refinement particularly helps with large degraded regions which are forced to implode gradually through the propagation of reliable outer pixels into the region.

The contributions of our approach are therefore as follows. We present a novel pixel-exemplar based restoration algorithm using spatiotemporal random walks. In comparison to current state-of-the-art archive film restoration techniques, our method is more accurate by using more reliable statistics produced during the random walks. Also, in addition to intensity and motion features, we employ a higher order texture feature, i.e. one that is more complex than raw intensities.

Finally, degraded pixels within a frame are collectively restored in a multiscale framework by updating all their features (intensity, motion and texture), which leads to more effective searching for optimal replacements (and significantly helps in the restoration of degraded regions that are considerably larger than typical defects). This means that at each scale the attributes of a defective pixel are updated with reference to the attributes of normal pixels and other defective pixels updated in the previous higher scale as long as they fall within the defective pixel's random walk-based spatiotemporal neighbourhood. Thus, there are more constraints to contribute to the restoration of intensities.

In Sect. 3, our proposed method is presented. First, the fundamentals of 3D random walks are introduced in Sect. 3.1. Then, our restoration algorithm is described in Sect. 3.2, and is followed by the multiscale restoration scheme which is briefly reviewed in Sect. 3.3. Finally in Sect. 4, we compare and evaluate our proposed method against two state-of-the-art methods, i.e. [15] and [10], on a variety of artificially degraded and real films.

## 2 Background

The task of filling in missing regions in single or consecutive frames is often referred to as *Inpainting*, which originates from restoration in the world of Art. It was first introduced into digital image restoration by Bertalmio et al. [1] who adapted the original idea of artistic inpainting by propagating the surrounding colour and structure into the missing area. Since then, inpainting has become a popular topic in computer vision and most of the research is concentrated on mainly two directions, i.e. image structure (non-texture) propagation methods and exemplar (texture) based methods.

Examples of methods developed to recover image structure information in degraded regions are [3,8] for edges and [18,4] for level lines. These usually require complex image models with high order partial differential equations or calculus of variations. Although such methods have proven to be effective solutions to restoring small gaps in degraded images, they suffer from blurring side-effects when dealing with large missing areas, e.g. they can fail to restore textural details within the missing regions they recover.

Exemplar based inpainting methods [2,5] attempt to overcome these side-effects. In a similar fashion to texture synthesis methods, e.g. [14,6], Criminisi et al. [5] performed the propagation of textures using a block-based sampling process, pointing out that the order of the filling process is critical for achieving simultaneous recovery of image structure and texture. Wexler et al. [25] and Patwardhan et al. [21] extended the algorithm in [5] by enlarging the sampling region to a number of temporal neighbouring frames (forwards and backwards) and both methods are designed to fill in space-time holes in video sequences with stationary background and moving foreground in periodic motions. Patwardhan et al. [21] further considered scenes with restricted camera motions by including a motion segmentation procedure. In [10], instead of using a global search

as performed in [25,21], Gangal et al. limited their search region to temporal motion compensated neighbourhoods. These three methods ([25], [21], and [10]) inherited the shortcomings of [5]: (a) it is difficult to choose the optimal size of an exemplar patch, considering that a larger patch will possibly bring artifacts while a smaller one may cause mismatching, and (b) a mismatching of patches in early stages will cause an incremental effect to the detriment of the final results.

We now focus on algorithms developed to restore missing scene information in archive films specifically. There is a class of methods that have used filter-based techniques applied to the entire image regardless of a defect map, e.g. the LUM filter [13], the ML3Dex filter [17] and the SMF filter [12]. These methods are able to go a long way in eliminating the defects but result in artifacts elsewhere in the image by removing texture detail. Recently, a series of methods [19,23,16,15] have applied statistical modelling to preform the defect detection and removal stages under a single framework. As the state-of-the-art, Kokaram's Bayesian framework [15] attempted to model noise and scratches, and perform motion adjustment together. Three binary variables were used for each pixel to mark if a pixel is degraded, forward occluded or backward occluded. These variables, together with restored image intensities and motion vectors, were defined as unknowns. Given the pixel values of degraded frames and initial motion estimations, a two-stage procedure was designed to estimate the variables and image intensities first and then adjust the motion vectors according to neighbouring motion vectors, before repeating this process for a fixed number of iterations. It is worth noting that in [15], to perform motion adjustment for a degraded pixel, the method relies on the accuracy of the pixel's surrounding motion vectors. During the iterative processing, motion information is improved separately and with no reference to the improved intensities. However as stated earlier, in our work, we update the motion vectors of a defective pixel in a multiscale process with reference to all attributes (i.e. intensity, motion and texture) of normal pixels and other defective pixels (updated in the higher scale) as long as they fall within its spatiotemporal random walk-based neighbourhood.

We assume defect maps  $\mathbf{D} = \{d_{\mathbf{x}}, d_{\mathbf{x}} \in \{0, 1\}\}$  for an archive film sequence are available using any reasonably accurate defect detection algorithm, e.g. the HAFID-STC defect detector proposed by Wang and Mirmehdi [24]. The label  $d_{\mathbf{x}} = 1$  states that pixel  $\mathbf{x}$  is degraded. The method proposed in [24] first trains a Hidden Markov Model (HMM) for defect-free temporal pixel sequences across a large number of frames which is then applied to unseen temporal pixel sequences to detect defects. However, this results in a considerable number of false alarms, which are then eliminated in [24] through a two-stage removal process based on (a) MRF modelling for false alarms that have strong correlation with their neighbours and (b) localised feature tracking for those that can be traced temporally. They achieved improved results in comparison to other techniques such as [15] and [22] and hence their approach is used to generate the input to our restoration process described here, although binary defect maps from any other technique will also be applicable.

### 3 Proposed Method

Traditional pixel-exemplar or patch-exemplar based restoration methods such as [7,25,10], search for the optimal exemplar amongst a square or rectangular region of pixels using sliding windows. A novel feature of our proposed method is that for each defective pixel examined, we explore a dynamically generated, random-walk based region of candidate pixel-exemplars to select the optimal replacement from. Every pixel in this region shares a significant similarity with the previous pixels in the region as defined by their features, i.e. intensity, motion and texture. A random walk starts from a degraded pixel and stops when it reaches a strong boundary in terms of a significant change in all the pixel features. The size of the region is thus determined on-the-fly and is based on the length of all the random walks (for the current defective pixel). We perform an empirically-determined fixed number of random walks for a degraded pixel to form a region (see Sect. 4 for details).

After building the region of candidate pixel-exemplars for a degraded pixel, we assign to each of them a likelihood of being the optimal replacement for the degraded pixel. This is obtained for each pixel-exemplar by first computing the average (geometric mean) of transition probabilities during each random walk which starts from the degraded pixel and visits the pixel-exemplar. Then the averaged probabilities from these random walks are summed up to get a measure of the similarity between the pixel-exemplar and the rest of the pixel-exemplars in the region (recalling that the transition probabilities are an indication of pixel similarities in a path). The higher this value, the higher is the similarity. This is then weighted by a reliability value, which measures the degree of degradation for each pixel-exemplar, to obtain its likelihood value. The pixel-exemplar with the maximal likelihood will be selected to replace the target degraded pixel. This means that the selected pixel is the optimal representation of the spatiotemporal random walk-based region of candidates - with relatively low (to possibly no) degree of degradation - to restore the current degraded pixel. The above processing is performed in multiscale for all degraded pixels within a frame along with their reliability values, refining the updated pixels' features from coarse to fine.

#### 3.1 Preliminaries and Definitions

Next, we state the fundamentals of a 3D random walk on an image sequence and then express the probability of a random walk sequence in the context of our application. We define the input image sequence as an undirected and weighted graph  $G = (V, E)$  with vertices (nodes)  $v_x \in V$  and edges  $e_{x', x''} \in E \subseteq V \times V$ . Each edge  $e_{x', x''}$  is assigned a weight  $w_{x', x''}$  where  $w_{x', x''} > 0$  and  $w_{x', x''} = w_{x'', x'}$ . An image pixel  $\mathbf{x}$  at location  $(i, j, t)$  ( $1 \leq i \leq Width, 1 \leq j \leq Height, 1 \leq t \leq Length$ ) is represented as a node  $v_x (v_x \in V)$  in graph  $G$  where  $Width \times Height \times Length$  defines the image sequence volume.

A random walk sequence  $Path_{0,K} = \{\mathbf{x}^0, \mathbf{x}^1, \dots, \mathbf{x}^K\}$  with length  $K + 1$  on graph  $G$  is specified as a sequence of nodes (pixels) which is a Markov process.

The probability of the transition  $p(\mathbf{x}^k|\mathbf{x}^{k-1})$  between consecutive pixels  $\mathbf{x}^{k-1}$  and  $\mathbf{x}^k$  is given as the weight  $w_{\mathbf{x}^{k-1},\mathbf{x}^k}$  on the edge  $e_{\mathbf{x}^{k-1},\mathbf{x}^k}$ . According to the Markov property of  $Path_{0,K}$ , the probability of a  $Path_{0,K}$  starting at pixel  $\mathbf{x}^0$  is defined as

$$p(Path_{0,K}) = \prod_{k=1}^K p(\mathbf{x}^k|\mathbf{x}^{k-1}) = \prod_{k=1}^K w_{\mathbf{x}^{k-1},\mathbf{x}^k}, \tag{1}$$

where  $P = \{Path_{0,K_m}^m\}_{m=1}^M$  is a set of  $M$  random walks on graph  $G$ , with each walk starting from  $\mathbf{x}^0$ . Furthermore, we define the region of candidates or pixel-exemplars  $\mathbf{R}_{\mathbf{x}^0} = \bigcup_{m=1}^M Path_{0,K_m}^m$  as the set of all pixels visited by the random walks in  $P$ . The neighbourhood for a pixel, the associated edge weights, and the walk length are expressed as follows:

*Neighborhood  $N_{\mathbf{x}}$* : For each pixel  $\mathbf{x}$  on a walk, we define a  $3 \times 3 \times 3$  spatiotemporal motion compensated neighbourhood  $N_{\mathbf{x}}$  centred at  $\mathbf{x}$ . In  $N_{\mathbf{x}}$ , we denote the connection between pixel  $\mathbf{x}$  and  $\mathbf{x}' (\mathbf{x}' \in N_{\mathbf{x}}, \mathbf{x}' \neq \mathbf{x})$  as edge  $e_{\mathbf{x},\mathbf{x}'}$  with a weight  $w_{\mathbf{x},\mathbf{x}'}$ . For each step in a random walk, a transition from the current pixel  $\mathbf{x}$  to one of its 26 direct neighbours  $\mathbf{x}' (\mathbf{x}' \in N_{\mathbf{x}})$  is permitted.

*Edge Weights*: In the same fashion as previous graph-based methods, e.g. [11], the edge weights are defined by a function that evaluates the similarity of two consecutive pixels during a random walk so as to bias it to stop the walk when a significant decrease in similarity is observed. Here, we define edge weights as the probability of pixels  $\mathbf{x}$  and  $\mathbf{x}'$  being identical, measured by using a number of different pixel features,

$$w_{\mathbf{x},\mathbf{x}'} = \frac{1}{T} \prod_{q=1}^Q \exp\left\{-\frac{\varphi_q^2(\mathbf{x}, \mathbf{x}')}{2\sigma_q^2}\right\}, \tag{2}$$

where  $T$  is a normalization constant,  $\sigma_q$  is the standard deviation for pixel feature  $q$ , and  $\varphi_q(\cdot)$  measures the Euclidean distance between pixel  $\mathbf{x}$  and  $\mathbf{x}'$  in feature space  $\mathfrak{F}_q$ . A variety of pixel features can be used to measure the similarity between two pixels and here we apply four (i.e.  $Q = 4$ ); these are intensity, forward and backward motion, and the local LBP texture pattern:

$$\varphi_q^2(\mathbf{x}, \mathbf{x}') = \frac{1}{J_q} \sum_{j=1}^{J_q} (Z_q^j(\mathbf{x}) - Z_q^j(\mathbf{x}'))^2, \tag{3}$$

where  $Z_q = \{\mathbf{I}, \mathbf{V}_{\mathbf{x}}^f, \mathbf{V}_{\mathbf{x}}^b, \mathbf{L}\}$  for  $q = \{1..4\}$ ,  $\mathbf{I}$  represents RGB intensity maps with  $J_1 = 3$ ,  $\mathbf{V}_{\mathbf{x}}^f$  and  $\mathbf{V}_{\mathbf{x}}^b$  represent forward and backward motion vector maps with  $J_2 = J_3 = 2$  respectively, and  $\mathbf{L}$  represents maps of 2D image LBP patterns in a spatial  $3 \times 3$  neighbourhood with  $J_4 = 8$ . The addition of a texture feature, along with a more integrated contribution of all the features used through (3), and subsequently (5), is an essential improvement on other works in archive film restoration, such as [15] and [10]. The extra texture feature is specifically appropriate to enforce a constraint in textured regions to help select the pixels that can be included in the region of candidates during the random walks.



*Walk Length.* We control the length of a random walk by monitoring  $p(\text{Path}_{0,K})$  in the same manner as proposed in [9]. Since we are performing a biased random walk by encouraging transitions between similar neighbours, the random walk will be terminated if  $p(\text{Path}_{0,K})$  is smaller than a threshold. This will prevent random walks from stepping across strong boundaries in terms of significant changes of all pixel features. A walk will also be terminated if it hits a hard boundary, i.e. the image boundaries on the spatial domain and the first and last frames on the temporal axis.

### 3.2 Restoration of Degraded Pixels

We restore all pixel features of a degraded pixel by replacing the degraded pixel with the optimal pixel-exemplar selected from its region of candidates, which has the maximal likelihood of being the original pixel. The selection procedure is as follows.

For each pixel-exemplar  $\mathbf{x}$  in a degraded pixel's region of candidates, i.e.  $\mathbf{R}_{\mathbf{x}^0}$ , the similarity between  $\mathbf{x}$  and the rest of the pixel-exemplars in the region is measured based on the probabilities of random walk paths which start from the degraded pixel and visit the pixel-exemplar, represented as

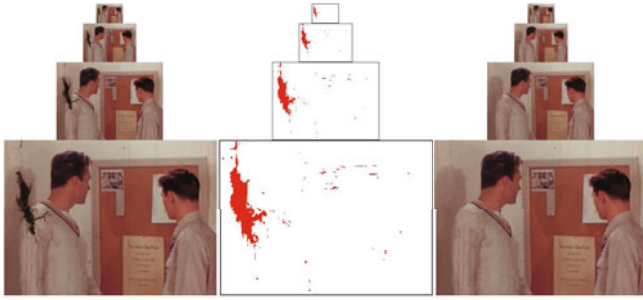
$$A_{\mathbf{x}} = \sum_{m=1}^M \sum_{k=1}^{K_m} \left( p(\text{Path}_{0,k}^m)^{1/k} \cdot \delta(\mathbf{x}^k = \mathbf{x}) \right), \quad (4)$$

where  $\delta(\cdot)$  is the Dirac *delta* function. In order to measure the similarity among all pixel-exemplars in a random walk path regardless of the length of the path, we compute the geometric mean of their transition probabilities. Provided we perform a sufficient number of spatiotemporal random walks, the sum of their averaged probabilities suggests the similarity between the pixel-exemplar and the rest of the pixel-exemplars in the region. This is influenced by the way spatial random walks are used in [9] to examine the transition probabilities (i.e. similarity) of pixels along a path in their image denoising application. The reason why we use this value instead of using other measurements, e.g. a count of random walks that visit the pixel-exemplar, is because this value indicates if the pixel-exemplar provides random walks with a smooth transition from their previous locations to this pixel-exemplar, e.g. if the probabilities of random walk paths decrease significantly after they visit this pixel-exemplar, this value will be probably small even though this pixel-exemplar has been visited by a large number of random walks. The optimal pixel-exemplar is then selected as

$$\hat{\mathbf{x}}^0 = \arg \max_{\mathbf{x} \in \mathbf{R}_{\mathbf{x}^0}} (A_{\mathbf{x}} \cdot \mathbf{r}(\mathbf{x})), \quad (5)$$

where  $\mathbf{r}(\cdot)$  indicates the reliability of a pixel-exemplar based on its degree of degradation. For normal pixels,  $\mathbf{r}(\cdot)$  is 1 while a degraded pixel is initialised to the likelihood of being identical to all its defect-free neighbours in  $N_{\mathbf{x}}$ :

$$\mathbf{r}(\mathbf{x}) = \begin{cases} \frac{1}{\sum_{\mathbf{x}' \in N_{\mathbf{x}}} \delta(d_{\mathbf{x}'})} \sum_{\mathbf{x}' \in N_{\mathbf{x}}} w_{\mathbf{x}\mathbf{x}'} \delta(d_{\mathbf{x}'}) & d_{\mathbf{x}} = 1 \\ 1 & d_{\mathbf{x}} = 0 \end{cases} \quad (6)$$



**Fig. 2.** (from left) A sample image pyramid, the defect map pyramid, and the restored results using the proposed method. The degraded regions are gradually recovered from coarse to fine and from the boundaries to their inner part.

Note also that by this definition, a false alarm pixel is more likely to be initialised with a high  $r(\cdot)$  value, given it is likely to be more similar to its defect-free spatiotemporal neighbours than to real degraded pixels. After a degraded pixel is replaced with a specific pixel-exemplar, its reliability value is updated with:

$$\hat{r}(\mathbf{x}) = \frac{1}{|N_{\mathbf{x}}|} \sum_{\mathbf{x}' \in N_{\mathbf{x}}} r(\mathbf{x}'). \tag{7}$$

During the multiscale updating algorithm (reviewed next), the  $r(\cdot)$  value for a degraded pixel will approach 1 after a number of updates. For a degraded pixel near the boundary of a degraded region, the  $r(\cdot)$  value will reach 1 faster than an inner pixel considering it is surrounded by more reliable spatiotemporal neighbours (normal pixels). Thus, during the multiscale refinement, a degraded region will gradually implode through the propagation of more reliable outer pixels in the region. For an example see Fig. 2.

Additionally, since a pixel in a false alarm region may be initialized with a larger  $r(\cdot)$  value (as noted above) than a real degraded pixel, then the false alarm pixel is more likely to obtain an optimal replacement considering more reliable candidates are present in its random walk-based neighbourhood.

### 3.3 Multiscale Refinement

Given an image sequence and its defect map, we build pyramids for each frame and its corresponding defect map by downsampling the original by a factor of 2 after smoothing with a  $5 \times 5$  Gaussian kernel. A sample image pyramid and its associated defect pyramid are shown in Fig. 2. After restoring the degraded pixels' features on a current level of the pyramid, we upsample these pixels to the next level and then update their corresponding pixels' features in that level. This level-by-level refinement and restoration process continues until it reaches the lowest level of the pyramid (see Algorithm 1).

**Algorithm 1.** The multiscale restoration algorithm

---

```

Build pyramids  $\{\mathbf{I}_s^f\}_{s=1}^S$  and  $\{\mathbf{D}_s\}_{s=1}^S$ ;
Initialize scale  $s = 1$ : Compute motion vector maps  $\mathbf{V}_1^f$ ,  $\mathbf{V}_1^b$ ,  $\mathbf{L}_1$ , and  $\mathbf{r}_1$ ;
while  $s \leq S$  do
  if  $s > 1$  then
    Update  $\mathbf{I}_s$ ,  $\mathbf{V}_s^f$ ,  $\mathbf{V}_s^b$ ,  $\mathbf{L}_s$  by  $\mathbf{I}_{s-1}$ ,  $\mathbf{V}_{s-1}^f$ ,  $\mathbf{V}_{s-1}^b$ ; /*Only on degraded sites*/
  end if
   $(\mathbf{I}_{s+1}, \mathbf{V}_{s+1}^f, \mathbf{V}_{s+1}^b, \mathbf{L}_{s+1}) = \text{Restoration}(\mathbf{I}_s, \mathbf{V}_s^f, \mathbf{V}_s^b, \mathbf{L}_s, \mathbf{D}_s)$ ;
  Update  $\mathbf{r}_{s+1}$  using equation (7);
  if  $s < S$  then
    Upsample  $\mathbf{I}_{s+1}, \mathbf{V}_{s+1}^f, \mathbf{V}_{s+1}^b, \mathbf{r}_{s+1}$  by factor 2; /*For every scale but the last*/
  end if
   $s = s + 1$ ;
end while

```

---

## 4 Experimental Results and Discussion

We present the restoration performance of the proposed algorithm on both artificially degraded and real sequences, and compare our results against two state-of-the-art techniques: Kokaram’s Bayesian framework [15] and Gangal and Dizardoglu’s exemplar-based method [10], hereafter referred to as Kokaram04 and GD06 respectively. The defect maps for both GD06 and the proposed method were produced in advance using the HAFID-STC defect detector [24] while Kokaram04 has an integrated defect detector. All methods were tuned for optimal performance using constant parameter values across all experiments.

**Synthetic defects** - The proposed method was compared against Kokaram04 and GD06 on restoring five artificially degraded real sequences totalling 1500 frames, namely *Mobile Calendar*, *Container*, *Foreman*, *News* and *Paris*. The degraded sequences were produced by adding synthetic black and white defects of sizes of between 1 and 6000 pixels on a random basis. For each method, the Mean Square Error (MSE) to measure the difference between the original defect-free frame  $F$  and the restored frame  $\hat{F}$  was computed:

$$\text{MSE}(F, \hat{F}) = \frac{1}{\text{Width} \times \text{Height} \times 3} \sum_{\mathbf{x} \in F} \sum_{i=1}^3 (F_{\mathbf{x}}^i - \hat{F}_{\mathbf{x}}^i)^2 \quad (8)$$

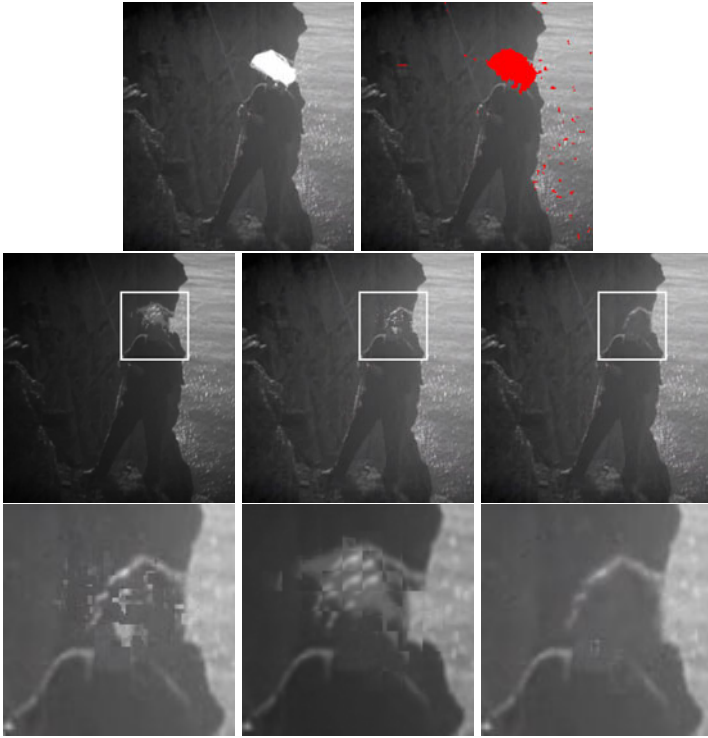
Columns 2 through to 5 in Table 1 show the MSEs for four randomly selected sample frames from the *Mobile Calendar*, *Container*, *Foreman*, *News* and *Paris* sequences respectively. The percentage of degraded pixels in each frame is listed along with the frame number along the top row. The raw, unrestored frame error rate is shown along the ‘Degraded’ row in each case. The last column in Table 1 shows the average MSEs across all frames in each of the synthetic-error sequences for each method; for example for the *Foreman* sequence, given the average true MSE rate of 153.4, the proposed method resulted in the lowest error at 44.7 compared to Kokaram04 and GD06 at 130.3 and 103.1 respectively. The proposed method performed much better in all the experiments, avoiding

the creation of too many artifacts (see e.g. Fig. 3) compared to Kokaram04 and GD06, and was also more capable of restoring large defects (e.g. Frame 233 of the *News* sequence). Note the introduction of artifacts during restoration by all three methods, may lead to MSE errors that are larger than the raw original whose MSE is only based on synthetic defects.

**Real defects** - We compared the three methods on restoring a variety of real degraded image sequences, including greyscale and colour, indoor and outdoor scenes, and slow and fast motions, and in all cases the proposed method produced the best results. In the following, three sets of sample results are illustrated to inspect three aspects of the proposed method, i.e. recovering a large degraded region and substantially avoiding artifacts in Fig. 3, handling defect-free (false alarm) pixels in Fig. 4, and correcting motions in Fig. 5.

**Table 1.** Comparison of MSEs on real sequences with synthetic errors

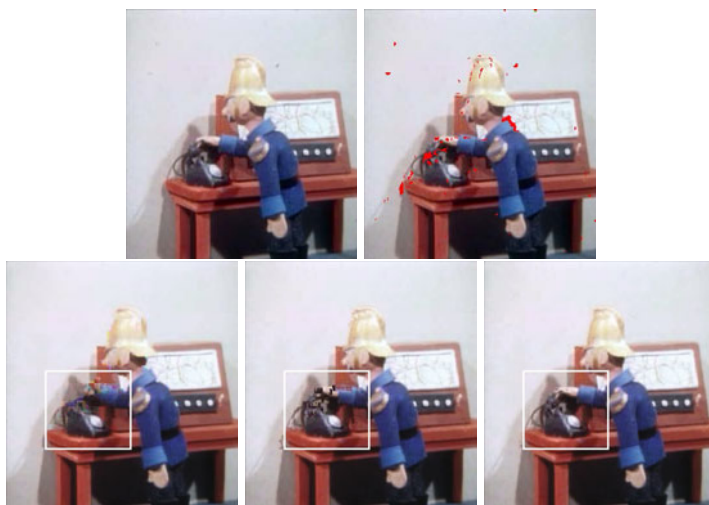
<i>Mobile Calendar</i>					
Frame #	32 (0.07%)	58 (0.11%)	181 (3.47%)	233 (0.03%)	Avg (0.62%)
Degraded	16.3	31.8	651.5	56.6	183.5
Kokaram04	210.6	89.9	293.6	92.5	157.3
GD06	128.7	80.7	196.8	123.5	135.9
Proposed	23.4	19.5	105.9	46.7	49.2
<i>Container</i>					
Frame #	9 (0.06%)	23 (0.21%)	138 (2.44%)	210 (0.06%)	Avg (0.60%)
Degraded	15.7	80.7	451.1	15.7	119.1
Kokaram04	7.6	3.5	93.4	2.8	33.9
GD06	0.9	0.8	96.3	3.2	23.8
Proposed	0.6	0.6	45.1	3.3	10.5
<i>Foreman</i>					
Frame #	33 (0.12%)	65 (0.24%)	98 (0.23%)	199 (0.02%)	Avg (0.55%)
Degraded	29.7	45.8	40.8	5.3	153.4
Kokaram04	70.5	155.32	89.84	74.0	130.3
GD06	95.3	149.12	95.87	68.5	103.1
Proposed	21.4	51.38	40.37	10.7	44.7
<i>News</i>					
Frame #	113 (0.08%)	165 (0.05%)	233 (2.47%)	291 (0.21%)	Avg (0.61%)
Degraded	31.5	15.2	751.5	66.9	154.5
Kokaram04	159.9	218.3	289.1	89.5	140.7
GD06	140.7	125.7	205.3	113.5	125.3
Proposed	49.5	27.7	119.8	54.4	54.2
<i>Paris</i>					
Frame #	81 (0.16%)	123 (0.01%)	158 (1.34%)	280 (0.10%)	Avg (0.60%)
Degraded	80.7	15.7	537.1	85.3	122.1
Kokaram04	30.5	25.6	107.1	11.3	63.9
GD06	21.8	27.9	122.3	15.4	83.8
Proposed	13.4	14.3	69.3	6.3	33.5



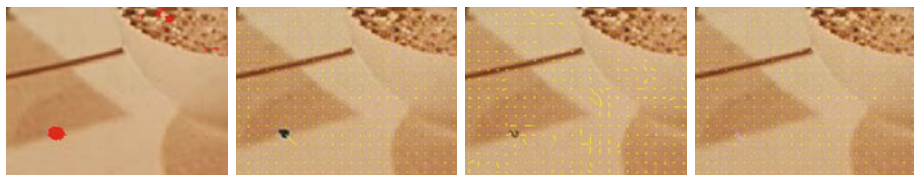
**Fig. 3.** *Cliff* - Comparing large missing area recovery. Top: original frame and the defect map in red; Middle: restoration results from Kokaram04, GD06, and the proposed method; Bottom: enlargement of selected areas.

Fig. 3 shows the results on a sample degraded frame with a large missing area. The original frame and the defect map (in red) are shown in the top row. The results by Kokaram04, GD06, and the proposed method are in the middle row and a close-up of the degraded area is shown in the bottom row. Kokaram04 results in a considerable number of artifacts in the restored frame because its performance strongly depends on the accuracy of motion information. Its motion correction procedure is not designed for such large missing areas, but rather for small degraded areas with accurate motion information provided in their spatial neighbouring regions. While GD06 is able to restore the outline of the man's head, it introduces some artifacts in the inner region due to the mismatching of patches in an early stage. Although the proposed method still causes some small artifacts, both the image structure and texture are recovered well.

In Fig. 4 we investigate the restoration performance of the three methods on handling false alarm pixels. A sample degraded frame and its corresponding defect map are in the top row, and restoration results from Kokaram04, GD06 and the proposed method follow in the bottom row. In this example all methods do well in restoring the real degraded pixels. However, both Kokaram04 and



**Fig. 4.** *Policeman* - Comparing restoration of false alarms. Top: original and its defect map; Bottom: restoration results from Kokaram04, GD06, and the proposed method.



**Fig. 5.** *Coffee* - Comparing motion correction. (from left) Original frame with overlaid defects, Original motion vectors, Corrected motion vectors from Kokaram04 and the proposed method.

GD06 lose considerable detail, e.g. the policeman's hand is missing in the frame restored by Kokaram04, and artifacts are introduced across the telephone and the policeman's hand in the frame restored by GD06.

The final example presents a comparison between Kokaram04 and the proposed method on correcting motions for degraded pixels. The motion vectors overlaid on an original frame are shown in the second image from left in Fig. 5. The correction results from Kokaram04's integrated motion correction algorithm and the proposed method follow this respectively. During Kokaram04's iterative process, motion information is improved separately and with no reference to the improved intensities; this means its correction is limited by the accuracy of initial motion estimations which are often inaccurate by the presence of defects. The proposed method outperforms Kokaram04 by achieving more accurate motion correction for each defective pixel by reference to their spatiotemporal random-walk neighbours through the multiscale process.

**Performance and Implementation Issues** - All methods were implemented in MATLAB on a laptop with Intel Core Duo 2.4 GHz and 2GB RAM. The average speed for a degraded frame of average size of  $480 \times 360$  was 406 seconds for our proposed method, while Kokaram04 and GD06 needed 174 and 265 seconds respectively. Our proposed algorithm is slower but more accurate than Kokaram04 and GD06, since it considers an extra feature and requires considerable sampling by the random walks. We experimented with different values of  $M$  by using different random seeds for each random walk.  $M = 800$  was found to provide stable results and reasonable computing costs. The number of steps in each random walk often varied from 2 to 48. Since accuracy is critical for the restoration of archive films, the extra computational burden is a tolerable cost.

## 5 Conclusion

We presented a novel pixel-exemplar based restoration algorithm using spatiotemporal random walks. The random walks are formed by considering pixel similarities using multiple features. The method is applicable given a defect map generated by any archive film defect detection algorithm. While the use of multiple features adds to our computational costs, we obtain much more accurate and artifact-free results than current state-of-the-art techniques.

**Acknowledgments.** We thank Great Western Research, ITV and Bristol University for funding this project.

## References

1. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proc. SIGGRAPH, pp. 417–424 (2000)
2. Bornard, R., Lecan, E., Laborelli, L., Chenot, J.H.: Missing data correction in still images and image sequences. In: Proc. MULTIMEDIA, pp. 355–361 (2002)
3. Chan, T.F., Kang, S.H., Shen, J.: Euler’s elastica and curvature driven diffusions. *SIAM J. App. Math.* 2, 564–592 (2002)
4. Chan, T.F., Shen, J.: *Image Processing and Analysis*. SIAM, Philadelphia (2005)
5. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. IP* 13(9), 1–13 (2004)
6. Efros, A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: Proc. SIGGRAPH, pp. 341–346 (2001)
7. Efros, A., Leung, T.: Texture synthesis by non-parametric sampling. In: Proc. ICCV, vol. 2, pp. 1033–1038 (1999)
8. Esedoglu, S., Shen, J.: Digital inpainting based on the mumford-shah-euler image model. *Euro. J. App. Math.* 13(4), 353–370 (2002)
9. Estrada, F., Fleet, D., Jepson, A.: Stochastic image denoising. In: Proc. BMVC (2009)
10. Gangal, A., Dizdaroglu, B.: Automatic restoration of old motion picture films using spatiotemporal exemplar-based inpainting. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2006. LNCS, vol. 4179, pp. 55–66. Springer, Heidelberg (2006)

11. Grady, L.: Random walk for image segmentation. *IEEE Trans. PAMI* 28(11), 1768–1783 (2006)
12. Hamid, M.S., Harvey, N., Marshall, S.: Genetic algorithm optimization of multi-dimensional grayscale soft morphological filters with applications in film archive restoration. *IEEE Trans. CSVT* 13(5), 406–416 (2003)
13. Hardie, R., Boncelet, C.: Lum filters: a class of rank-order-based filters for smoothing and sharpening. *IEEE Trans. SP* 41(3), 1061–1076 (1993)
14. Heeger, D.J., Bergen, J.R.: Pyramid-based texture analysis/synthesis. In: *Proc. SIGGRAPH*, pp. 229–238 (1995)
15. Kokaram, A.: On missing data treatment for degraded video and film archives: a survey and a new bayesian approach. *IEEE Trans. IP* 13(3), 397–415 (2004)
16. Kokaram, A., Godsill, S.: Mcmc for joint noise reduction and missing data treatment in degraded video. *IEEE Trans. SP* 50(2), 189–205 (2002)
17. Kokaram, A., Morris, R., Fitzgerald, W., Rayner, P.: Interpolation of missing data in image sequences. *IEEE Trans. IP* 4(11), 1509–1519 (1995)
18. Masnou, S., Morel, J.M.: Level lines based disocclusion. In: *Proc. ICIP*, vol. 3, pp. 259–263 (1998)
19. Morris, R.: Image Sequence Restoration using Gibbs Distributions. Ph.D. thesis, Cambridge University (1995)
20. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE trans. PAMI* 24(7), 971–987 (2002)
21. Patwardhan, K., Sapiro, G., Bertalmio, M.: Video inpainting of occluding and occluded objects. In: *Proc. ICIP*, vol. 2, pp. II: 69–72 (2005)
22. Ren, J., Vlachos, T.: Efficient detection of temporally impulsive dirt impairments in archived films. *Signal Processing* 87(3), 541–551 (2007)
23. Roosmalen, P.M.B.V.: Restoration of Archived Film and Video. Ph.D. thesis, Delft University of Technology (1999)
24. Wang, X., Mirmehdi, M.: HMM based archive film defect detection with spatial and temporal constraints. In: *Proc. BMVC* (2009)
25. Wexler, Y., Shechtman, E., Irani, M.: Space-time video completion. In: *Proc. CVPR*, vol. 1, pp. 120–127 (2004)



# Reweighted Random Walks for Graph Matching

Minsu Cho, Jungmin Lee, and Kyoung Mu Lee

Department of EECS, ASRI, Seoul National University, 151-742, Seoul, Korea

<http://cv.snu.ac.kr>

**Abstract.** Graph matching is an essential problem in computer vision and machine learning. In this paper, we introduce a random walk view on the problem and propose a robust graph matching algorithm against outliers and deformation. Matching between two graphs is formulated as node selection on an association graph whose nodes represent candidate correspondences between the two graphs. The solution is obtained by simulating random walks with reweighting jumps enforcing the matching constraints on the association graph. Our algorithm achieves noise-robust graph matching by iteratively updating and exploiting the confidences of candidate correspondences. In a practical sense, our work is of particular importance since the real-world matching problem is made difficult by the presence of noise and outliers. Extensive and comparative experiments demonstrate that it outperforms the state-of-the-art graph matching algorithms especially in the presence of outliers and deformation.

**Keywords:** graph matching, random walks, feature correspondence.

## 1 Introduction

Graph matching is an essential problem in theoretical computer science; it is related to various research areas in computer vision, pattern recognition, and machine learning [1]. The problem of graph matching is to determine a mapping between the nodes of the two graphs that preserves the relationships between the nodes as much as possible. In computer vision, it is widely known that the fundamental problem of establishing correspondences between two sets of visual features can be effectively solved by graph matching. Thus, graph matching is used in various tasks, such as feature tracking, image retrieval, object recognition, and shape matching. Many graph matching algorithms proposed in the 1980s and 1990s focused on exploiting relatively weak unary and pair-wise attributes and did not specifically aim at optimizing a well-defined objective function [1]. Recent resurgence of combinatorial optimization approaches to feature matching [2,3,4,5,6,7,8] has changed the situation and firmly settled graph matching formulations based on Integer Quadratic Programming (IQP), which is a generalization of the classical graph matching problems. IQP explicitly takes into consideration both unary and pair-wise terms reflecting the compatibilities in local appearance as well as the pair-wise geometric relationships between the matching features. Since IQP is known to be NP-hard, approximate solutions

are required. Our work provides a novel interpretation of graph matching in a random walk view and relates it to the IQP formulation. Introducing an association graph constructed with nodes as candidate correspondences and edges as pair-wise compatibilities between candidate correspondences, we show that the search for correspondences between the given two graphs can be cast as a node ranking [9,10] and selection problem in the association graph. For this ranking, we introduce an *affinity-preserving random walk* and derive a ranking based on its quasi-stationary distribution, and prove its equivalence to the spectral relaxation [3] for the IQP formulation. Then, in this random walk view, we adopt the personalization strategy of Web ranking algorithms [11] and propose the *reweighted random walk* algorithm by reweighting jumps for the graph matching constraints. It achieves noise-robust graph matching by simultaneously updating and exploiting the confidences of candidate correspondences. In a practical sense, our work is of particular importance since the real-world matching problem is made difficult by the presence of deformation and outliers.

A myriad of algorithms have been proposed for graph matching, and those closely related to ours are follows. Maciel and Costeira [12] formulated graph matching as a constrained integer optimization problem with a concave optimization scheme, but the complexity of its minimization was still non-polynomial. Gold and Rangarajan [13] proposed the Graduated Assignment (GAGM) algorithm to solve the IQP by relaxing the integer constraint. In their deterministic annealing approach, GAGM gradually updates the derivative of the relaxed IQP in the soft assignment step driven by an annealing schedule. The SPGM algorithm proposed by van Wyk and van Wyk [14] iteratively updates the objective function of IQP by projecting an approximation of the current matching matrix onto the convex space of the matching constraints. Leordeanu and Hebert [3] proposed a simple and efficient approximation to the IQP using spectral relaxation, which computes the leading eigenvector of symmetric nonnegative affinity matrix. Their Spectral Matching (SM) ignored the integer constraints in the relaxation step and induced them during the discretization step by a greedy approach. They also recently proposed an iterative matching method (IPFP) [8] with climbing and convergence properties which optimizes the IQP in the discrete domain. Cour et al. [4] extended SM [3] to Spectral Matching with Affine Constraint (SMAC) by introducing affine constraints into the spectral decomposition that encodes the one-to-one matching constraints. Lee et al. [15] presented a Markov chain Monte Carlo algorithm to solve the IQP based on the spectral relaxation. Zass and Shashua [6] showed that matching problems can be represented by a matrix constructed by Kronecker products, and also introduced a probabilistic framework for hypergraph matching. Duchenne et al. [7] extended the method of [3] to high-order graph matching which was formulated as a tensor eigendecomposition problem. Our problem formulation is related to the previous IQP formulations of [13,3,4,15], but we approached it from a random walk view. Note that the previous random walk-based approaches [16,17] use the random walk theory to find a signature for each node in a graph, and their problem formulations are different from ours.

This paper presents three main contributions. First, it establishes a novel random walk view for graph matching and provides a basis for random walk interpretations of recent spectral matching [3,4,7] and other iterative algorithms [13,6]. Second, in this view, we propose a powerful matching algorithm inspired by the personalization strategy of Web ranking algorithms [11] and the Sinkhorn method [18]. Third, it is extensively demonstrated against several state-of-the-arts graph matching algorithms. The comparison not only reveals the superior performance of our algorithm but also facilitates a comprehensive study of recent graph matching algorithms.

## 2 Problem Formulation

The objective of graph matching is to determine the correct correspondences between two attributed graphs  $G^P = (V^P, E^P, A^P)$  and  $G^Q = (V^Q, E^Q, A^Q)$ , where  $V$  represents a set of nodes,  $E$ , edges, and  $A$ , attributes. Each node  $v_i^P \in V^P$  or edge  $e_{ij}^P \in E^P$  has an associated attribute vector  $\mathbf{a}_i^P \in A^P$  or  $\mathbf{a}_{ij}^P \in A^P$ . In feature correspondence problems, a node attribute  $\mathbf{a}_i^P$  usually describes a local appearance of feature  $i$  in an image  $P$ , and an edge attribute  $\mathbf{a}_{ij}^P$  represents the geometric relationship between features  $i$  and  $j$  in the image  $P$ . For each pair of edges  $e_{ij}^P \in E^P$  and  $e_{ab}^Q \in E^Q$ , there is an affinity or compatibility  $\mathbf{W}_{ia:jb} = f(\mathbf{a}_i^P, \mathbf{a}_j^P, \mathbf{a}_{ij}^P, \mathbf{a}_a^Q, \mathbf{a}_b^Q, \mathbf{a}_{ab}^Q)$  that measures the mutual consistency of attributes between the pairs of candidate correspondences  $(v_i^P, v_a^Q)$  and  $(v_j^P, v_b^Q)$ . Thus, using a matrix form  $\mathbf{W}$ , a non-diagonal element  $\mathbf{W}_{ia:jb}$  contains a pair-wise affinity between two correspondences  $(v_i^P, v_a^Q)$  and  $(v_j^P, v_b^Q)$ , and a diagonal term  $\mathbf{W}_{ia:ia}$  represents a unary affinity of a correspondence  $(v_i^P, v_a^Q)$ . Representing the correspondence with an assignment or permutation matrix  $\mathbf{X} \in \{0, 1\}^{n^P \times n^Q}$  is common, such that  $\mathbf{X}_{ia} = 1$  implies that node  $v_i^P$  corresponds to node  $v_a^Q$ , e.g., feature  $i$  in the image  $P$  is matched to feature  $a$  in the image  $Q$ , and  $\mathbf{X}_{ia} = 0$  otherwise. In this paper, we denote  $\mathbf{x} \in \{0, 1\}^{n^P n^Q}$  as a column-wise vectorized replica of  $\mathbf{X}$ . The graph matching problem can be formulated as an integer quadratic program (IQP), that is, finding the indicator vector  $\mathbf{x}^*$  that maximizes the quadratic score function as follows.

$$\mathbf{x}^* = \arg \max(\mathbf{x}^T \mathbf{W} \mathbf{x}) \tag{1}$$

$$s.t. \ \mathbf{x} \in \{0, 1\}^{n^P n^Q}, \ \forall i \sum_{a=1}^{n^Q} \mathbf{x}_{ia} \leq 1, \ \forall a \sum_{i=1}^{n^P} \mathbf{x}_{ia} \leq 1,$$

where the two-way constraints refer to the one-to-one matching from  $G^P$  to  $G^Q$ . In general, no efficient algorithm exists that can guarantee the optimality bounds since the IQP is NP-hard, thus approximate solutions are required.

### 3 Random Walks for Graph Matching

Basically, the problem of graph matching between the two graphs  $G^P$  and  $G^Q$  can be interpreted in a random walk view by constructing an association graph  $G^{rw} = (V^{rw}, E^{rw}, A^{rw})$  as follows. Given the pair-wise affinity matrix  $\mathbf{W}$ , we consider each candidate correspondence  $(v_i^P, v_a^Q) \in V^P \times V^Q$  as a node  $v_{ia} \in V^{rw}$ , its associated weight  $\mathbf{W}_{ia;jb}$  as the attribute  $\mathbf{a}_{ia;jb} \in A^{rw}$  of the edge  $e_{ia;jb} \in E^{rw}$ . This is illustrated by an example in Fig.1(a). The original graph matching problem between  $G^P$  and  $G^Q$  is equivalent to selecting reliable nodes in the graph  $G^{rw}$  since the selected nodes in  $G^{rw}$  corresponds to graph or subgraph matching between  $G^P$  and  $G^Q$ . To select the nodes in  $G^{rw}$ , we adopt the statistics of the Markov random walks which has been used to compute the ranking or relevance of graphs in the Web environments [9],[10]. Thus, graph matching between  $G^P$  and  $G^Q$  can be transformed into the node ranking and selection problem by random walks on  $G^{rw}$ . In this view, we introduce an *affinity-preserving random walk* algorithm in Sec.3.1, which paves the way for *reweighted random walk* algorithm in Sec.3.2.

#### 3.1 Affinity-Preserving Random Walks

The standard way to define a random walk on a graph is to allow a random walker to take off on an arbitrary node and then successively visit new nodes by randomly selecting one of the outgoing edges according to a Markov transition kernel of the graph. In general, in order to define the transition matrix on weighted graphs, traditional random walk approaches convert affinity or weight matrix  $\mathbf{W}$  to the row stochastic matrix by  $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$ , where  $\mathbf{D}$  is a diagonal matrix with entries  $\mathbf{D}_{ii} = d_i = \sum_j \mathbf{W}_{ij}$ . This normalization is required not only for transforming  $\mathbf{W}$  into a stochastic matrix, that is “stochasticizing”, but also for other particular reasons in the applications. For example, in PageRank [10], each out-going hyperlink from a node  $i$  is row-normalized by  $1/d_i$  so that every webpage has the same total out-going weights. We may state this idea as

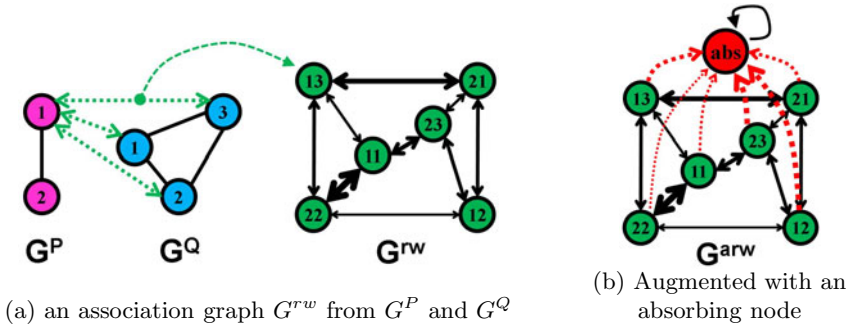


Fig. 1. Association Graphs for Graph Matching by Random Walks

Internet Democracy: each webpage has a total of one vote [19]. However, this democratic normalization is problematic in our approach for graph matching since in  $G^{rw}$  some nodes correspond to false candidate correspondences (outlier nodes) or more distorted ones than others. In such cases, the normalization can strengthen the adverse effect of outliers and weak correspondences, and pervert random walkers. For an example, consider Fig 1(a) where  $v_1^P$  and  $v_2^P$  correspond to  $v_1^Q$  and  $v_2^Q$ , respectively. The democratic normalization on  $G^{rw}$  scales up the affinities of outgoing edges of outlier nodes such as  $v_{12}$ ,  $v_{13}$ ,  $v_{21}$ , and  $v_{23}$  compared with the affinities of two inlier nodes  $v_{11}$  and  $v_{22}$  because the affinity sum of an outlier node is usually smaller than that of an inlier node.

How then can we preserve the original affinity relations while transforming the affinity matrix into the stochastic transition matrix for random walks? We define the maximum degree  $d_{\max} = \max_i d_i$ , and construct an augmented graph  $G^{arw}$  with an absorbing node  $v_{\text{abs}}$  which soaks affinity  $d_{\max} - d_i$  out of all the nodes  $v_i \in V^{rw}$  as shown in Fig 1(b). We treat this graph as a special Markov chain which has an absorbing node, i.e., a state which, once reached, cannot be transitioned out of. Since each node in the affinity matrix of  $G^{arw}$  has the same degree of  $d_{\max}$ , its normalized affinity matrix by  $1/d_{\max}$  results in a stochastic matrix and corresponds to an *absorbing Markov chain* [20] which preserves the relative affinity relations of the original graph  $G^{rw}$ . We call this approach an “affinity-preserving random walk”, and formulate its transition matrix  $\mathbf{P}$  and absorbing Markov chain as follows.

$$\mathbf{P} = \begin{pmatrix} \mathbf{W}/d_{\max} & \mathbf{1} - \mathbf{d}/d_{\max} \\ \mathbf{0}^T & 1 \end{pmatrix}, \quad \begin{pmatrix} \mathbf{x}^{(n+1)T} & x_{\text{abs}}^{(n+1)} \end{pmatrix} = \begin{pmatrix} \mathbf{x}^{(n)T} & x_{\text{abs}}^{(n)} \end{pmatrix} \mathbf{P}, \quad (2)$$

where  $\mathbf{W}/d_{\max}$  is the  $n^P n^Q \times n^P n^Q$  substochastic matrix, and  $\mathbf{1}$  is a  $n^P n^Q \times 1$  vector with all elements 1, and  $\mathbf{0}$  with all elements 0. This absorbing Markov chain has transient nodes of  $V^{rw}$  from which its random walker is certain to be absorbed into an absorbing node  $v_{\text{abs}}$ . Its steady state distribution is always  $(\mathbf{0}^T \ 1)$ , thus cannot be used for node ranking in the same way as PageRank [10]. For ranking on the absorbing Markov chain, we denote  $X^{(n)}$  as the node where a random walker in the absorbing Markov chain of Eq. (2) stays at time  $n$ , and define the conditional distribution  $\bar{\mathbf{x}}^{(n)}$  as

$$\bar{\mathbf{x}}_{ia}^{(n)} = P(X^{(n)} = v_{ia} \mid X^{(n)} \neq v_{\text{abs}}) = \frac{\mathbf{x}_{ia}^{(n)}}{1 - x_{\text{abs}}^{(n)}}, \quad (3)$$

which refers to the distribution of unabsorbed random walkers at time  $n$ . If  $\bar{\mathbf{x}}^{(n+1)} = \bar{\mathbf{x}}^{(n)} = \bar{\mathbf{x}}$ , we call  $\bar{\mathbf{x}}$  a *quasi-stationary distribution* of the absorbing Markov chain. This corresponds to a steady-state distribution in the Markov chain without absorbing nodes. Following the approach of PageRank [10] based on the steady-state distribution in ergodic Markov chains, we define the affinity-preserving PageRank as follows.

**Definition 1.** *The affinity-preserving PageRank of a graph with affinity matrix  $\mathbf{W}$  is the quasi-stationary probability  $\bar{\mathbf{x}}$  of Eq. (3) in affinity-preserving random walks of Eq. (2).*

**Theorem 1.** *The quasi-stationary distribution of the affinity-preserving random walks of Eq.(2) is proportional to the left principal eigenvector of  $\mathbf{W}$ .*

*Proof.* According to Eq.(2),

$$\begin{pmatrix} \mathbf{x}^{(n+1)T} & x_{\text{abs}}^{(n+1)} \end{pmatrix} = \begin{pmatrix} \mathbf{x}^{(n)T} \mathbf{W} / d_{\text{max}} & 1 - \mathbf{x}^{(n)T} \mathbf{d} / d_{\text{max}} \end{pmatrix}$$

Thus, by Eq.(3),

$$\bar{\mathbf{x}}^{(n+1)} = \frac{\mathbf{x}^{(n)T} \mathbf{W}}{\mathbf{x}^{(n)T} \mathbf{d}} = \frac{\bar{\mathbf{x}}^{(n)T} \mathbf{W}}{\bar{\mathbf{x}}^{(n)T} \mathbf{d}}$$

Then, since  $\bar{\mathbf{x}}^{(n+1)} = \bar{\mathbf{x}}^{(n)} = \bar{\mathbf{x}}$  for quasi-stationary distribution,  $\bar{\mathbf{x}}$  must satisfy  $\lambda \bar{\mathbf{x}}^T = \bar{\mathbf{x}}^T \mathbf{W}$ . If  $\mathbf{W}$  is irreducible, and the elements of  $\bar{\mathbf{x}}$  are non-negative, then it follows from the extended Perron Frobenius theorem that  $\lambda$  is the real maximal eigenvalue of  $\mathbf{W}$ , and  $\bar{\mathbf{x}}$  is a normalized non-negative left eigenvector of  $\mathbf{W}$  corresponding to the maximal eigenvalue. Therefore, the quasi-stationary distribution  $\bar{\mathbf{x}}$  is equivalent to the left eigenvector of  $\mathbf{W}$  with non-negative components.

Interestingly, this affinity-preserving PageRank is the solution of a relaxed version of the original IQP problem, and is equivalent to the spectral relaxation of [3]. By dropping two-way matching constraints and relaxing integer constraints from Eq.(1), the original IQP is approximated to a continuous problem as

$$\mathbf{x}^* = \arg \max(\mathbf{x}^T \mathbf{W} \mathbf{x}) \text{ s.t. } \mathbf{x} \in [0, 1]^{n^P n^Q}, \tag{4}$$

which is interpreted as a classical Rayleigh quotient problem in [3], whose solution  $\mathbf{x}^*$  is obtained by the eigenvector associated with the largest eigenvalue of  $\mathbf{W}$ . The result is the same as the affinity-preserving PageRank in our random walk view, and can also be computed efficiently by the power iteration method. This view provides a basis for random walk interpretations of recent spectral methods [3,4,7] and iterative algorithms [13,6] on graph matching problem. Assuming that the solution of the relaxed problem is close to the optimal discrete solution, the final solution is obtained by incorporating the matching constraints on it. A greedy mapping [3] or the Hungarian algorithm [21] can be adopted for the final discretization.

As demonstrated in our experiment in Sec.4, the affinity-preserving random walk matching (equivalent to SM in Sec.4) consistently outperforms conventional random walk matching (denoted by NRW in Sec.4) which uses row-normalized affinity matrix.

### 3.2 Reweighted Random Walks

In the previous affinity-preserving random walks, the matching constraints of Eq.(1) are ignored and not reflected in the random walk process. Inducing the matching constraints only as a post-processing discretization step like [3] leads to a weak local optimum. How then can we reflect the two-way matching constraints in the affinity-preserving random walk? We adopts the personalization

approach widely used in Web ranking methods [11,19], which strengthens the effects of reliable nodes in random walks. This is achieved by adopting a jump or teleport in the random walk: the random walker moves by traversing an edge with probability  $\alpha$  or by performing a jump to some constrained nodes with probability  $1 - \alpha$ .  $\alpha$  represents the bias between the two possible actions, i.e., following an edge or jumping. To address the lack of personalization or user’s focus, Web ranking algorithms adopted this approach in topic-sensitive or query-dependent variants [19]. In our formulation, adopting the personalized jump, the probability distribution is updated using the following equation:

$$\begin{pmatrix} \mathbf{x}^{(n+1)T} & x_{\text{abs}}^{(n+1)} \end{pmatrix} = \alpha \begin{pmatrix} \mathbf{x}^{(n)T} & x_{\text{abs}}^{(n)} \end{pmatrix} \mathbf{P} + (1 - \alpha) \mathbf{r}^T, \quad (5)$$

where a *reweighting jump* vector  $\mathbf{r}$  is added to the affinity-preserving random walk of Eq. (2). In this approach, we use the jumps for generating a biased random walk to the matching constraints. One possible way is to use the result of the discrete assignment mapping of current  $\mathbf{x}$  as the jump vector  $\mathbf{r}$  at each iteration. However, this scheme is vulnerable to the discretization of the wrong solution in early steps. Thus, we propose a robust reweighting scheme as described in Algorithm.1. The reweighting procedure consists of two steps: inflation and bistochastic normalization [18]. The inflation step of  $\exp(\beta \mathbf{x} / \max \mathbf{x})$  attenuates small values of  $\mathbf{x}$  and amplifies large values of  $\mathbf{x}$ . In this way, unreliable correspondences contribute insignificantly through the individual exponentials over the components of  $\mathbf{x}$ . Then, for the two-way constraint that a node in the graph  $G^P$  must correspond to only one node in the graph  $G^Q$  and vice versa, the bistochastic normalization scheme of Sinkhorn [18] is applied as in [13], which alternatively normalizes the rows and columns of  $\mathbf{X}$  (matrix form of  $\mathbf{x}$ ). Any square matrix whose elements are all positive is proven to converge to a bistochastic matrix [4] just by the iterative process [18]. At each iteration, the reweighting jumps are introduced on the transient part of the current affinity-preserving random walking  $\mathbf{x}^{(n)T} \mathbf{W} / d_{\max}$ . Thus, the reweighted random walk is formulated by

$$\begin{pmatrix} \mathbf{x}^{(n+1)T} & x_{\text{abs}}^{(n+1)} \end{pmatrix} = \alpha \begin{pmatrix} \mathbf{x}^{(n)T} & x_{\text{abs}}^{(n)} \end{pmatrix} \mathbf{P} + (1 - \alpha) \begin{pmatrix} f_C(\mathbf{x}^{(n)T} \mathbf{W})^T & 0 \end{pmatrix}, \quad (6)$$

where  $f_C(\cdot)$  denotes the reweighting function incorporating two-way constraints. Note that this is a dynamic Markov chain whose jump distribution is dynamically varying and dependent on the present distribution of  $\mathbf{x}$  unlike conventional jumps in random walks [11,19]. As the  $f_C(\cdot)$  generates a jump distribution close to a good solution, the subsequent random walks strengthen the distribution and move toward a integer solution. Its fast convergence is observed empirically in all our experiments. To further tighten the random walks by the matching constraints, we enforce *conflicting walk prevention* which entails that random walks to conflicting nodes are prevented according to the matching constraints. In the

---

<sup>1</sup> A bistochastic matrix is a matrix whose elements are all positive and whose rows and columns all add up to one: it may roughly be thought of as the continuous analog of a permutation matrix allowing  $x_i \in [0, 1]$ .

---

**Algorithm 1.** Reweighted Random Walk Graph Matching

---

- 1: Given the weight matrix  $\mathbf{W}$ , the reweight factor  $\alpha$ , and the inflation factor  $\beta$
  - 2: Prevent conflicting walks by setting  $\mathbf{W}_{ia;jb} = 0$  for all conflicting match pairs
  - 3: Set the maximum degree  $d_{\max} = \max_{ia} \sum_{jb} \mathbf{W}_{ia;jb}$
  - 4: Initialize the transition matrix  $\mathbf{P} = \mathbf{W}/d_{\max}$ , the starting probability  $\mathbf{x}$  as uniform
  - 5: **repeat**
  - 6:   ( Affinity-preserving random walking by edges )
  - 7:    $\bar{\mathbf{x}}^T = \mathbf{x}^T \mathbf{P}$
  - 8:   ( Reweighting with two-way constraints )
  - 9:    $\mathbf{y}^T = \exp(\beta \bar{\mathbf{x}} / \max \bar{\mathbf{x}})$
  - 10:   **repeat**
  - 11:     normalize across rows by  $\mathbf{y}_{ai} = \mathbf{y}_{ai} / \sum_{i=1}^I \mathbf{y}_{ai}$
  - 12:     normalize across columns by  $\mathbf{y}_{ai} = \mathbf{y}_{ai} / \sum_{a=1}^A \mathbf{y}_{ai}$
  - 13:     **until**  $\mathbf{y}$  converges
  - 14:      $\mathbf{y} = \mathbf{y} / \sum \mathbf{y}_{ai}$
  - 15:     ( Affinity-preserving random walking with reweighted jumps)
  - 16:      $\mathbf{x}^T = \alpha \bar{\mathbf{x}}^T + (1 - \alpha) \mathbf{y}^T$
  - 17:      $\mathbf{x} = \mathbf{x} / \sum \mathbf{x}_{ai}$
  - 18:     **until**  $\mathbf{x}$  converges
  - 19: Discretize  $\mathbf{x}$  by the matching constraints
- 

case of two-way constraints of Eq.(10), a random walker in node  $v_{ia}$  is prohibited to move to nodes  $\forall b \neq a, v_{ib}$  and  $\forall j \neq i, v_{ja}$ . This is easily implemented by initially eliminating such conflicting elements in the affinity matrix  $\mathbf{W}$ .

The quasi-stationary distribution of this reweighted random walk is efficiently computed using the power iteration method as summarized in Algorithm.1. Its computational complexity is  $O(|E^P||E^Q|)$  per iteration, where  $|E^P|$  and  $|E^Q|$  are the numbers of edges in the two graphs, respectively. In the final discretization step, any linear assignment algorithm can be adopted, such as a greedy algorithm in [3] or the Hungarian algorithm [21].

From an algorithmic point of view, our method has some resemblance to SM [3] and GAGM [13]. Without the reweighting jumps, our affinity-preserving random walking can be considered as the power iteration version of [3] as explained in Sec.3.1. The Sinkhorn method [18] introduced in our reweighting jumps is also adopted in the softassign step of GAGM [13]. However, our reweighting step does not require a deterministic annealing schedule as GAGM and is designed to effectively select reliable nodes for reweighted jumps. Our method provides faster convergence to a better optimum by the balance of walks and jumps as demonstrated in the experiments.

## 4 Experiments

We performed intensive experiments for the proposed method in three tasks: (1) synthetically generated random graphs, (2) point matching task using the CMU House image sequence [2], and (3) feature matching using real images. These three

---

<sup>2</sup> <http://vasc.ri.cmu.edu/idb/html/motion/>



experiments are designed to evaluate the performance of our algorithm (RRWM) on various graph matching tasks and to compare it with other state-of-the-art methods: SM[3], SMAC[4], HGM[6], IPFP[8], GAGM[13], and SPGM[14]. We additionally tested the performance of the random walk matching with conventional row-wise normalization denoted by NRWM. For SMAC[3] and HGM[4], the publicly available codes by authors were used, and SM, IPFP, and SPGM were implemented by us. For GAGM, we revised and tuned the code based on the implementation provided by Cour[4]. All methods were implemented using MATLAB and tested on 2.40 GHz Core2 Quad desktop PC. For each trial in all experiments, the same affinity matrix was shared as the input[5] and the Hungarian algorithm[6] was commonly used at final discretization step for all methods. Control parameters of GAGM and SPGM were based on the authors' papers and tuned for better performance. For our RRWM, we fixed  $\alpha = 0.2$ ,  $\beta = 30$  in all experiments. These settings allow us to quantify the accuracy and robustness of all algorithms, and fairly compare them with one another.

#### 4.1 Synthetic Random Graph Matching

In this experiment, following the experimental protocol of [13,4], we performed a comparative evaluation on random graph matching problems. For each trial, we constructed two graphs,  $G^P$  with  $n^P = n_{in} + n_{out}^P$  nodes and  $G^Q$  with  $n^Q = n_{in} + n_{out}^Q$  nodes, each consisting of  $n_{in}$  inlier nodes and the other outlier nodes. The reference graph  $G^P$  is generated with random edges of edge density  $\rho$ , where each edge  $e_{ij}^P \in E^P$  was assigned a random attribute  $\mathbf{a}_{ij}^P$  distributed uniformly in  $[0, 1]$ . We then created a perturbed graph  $G^Q$  by adding noise on the edge attributes between inlier nodes:  $\mathbf{a}_{ab}^Q = \mathbf{a}_{p(i)p(j)}^P + \varepsilon$ , where  $p(\cdot)$  is a random permutation function for inlier nodes. The deformation noise  $\varepsilon$  was distributed using the Gaussian noise function  $N(0, \sigma^2)$ . All the other edges connecting at least one of the outlier nodes are randomly generated as the same way in  $G^P$ . Thus, two graphs  $G^P$  and  $G^Q$  have a common and perturbed subgraph with size  $n_{in}$ . The affinity matrix  $\mathbf{W}$  was computed by  $W_{ia,jb} = \exp(-|\mathbf{a}_{ij}^P - \mathbf{a}_{ab}^Q|^2 / \sigma_s^2)$ ,  $\forall e_{ij}^P \in E^P, \forall e_{ab}^Q \in E^Q$ . The scaling factor  $\sigma_s^2$  is set to 0.15 empirically to show the best average performance of all the methods. The accuracy is measured by the number of detected true matches divided by the total number of ground truths, and the objective score by computing  $\mathbf{x}^T \mathbf{W} \mathbf{x}$  of the IQP objective.

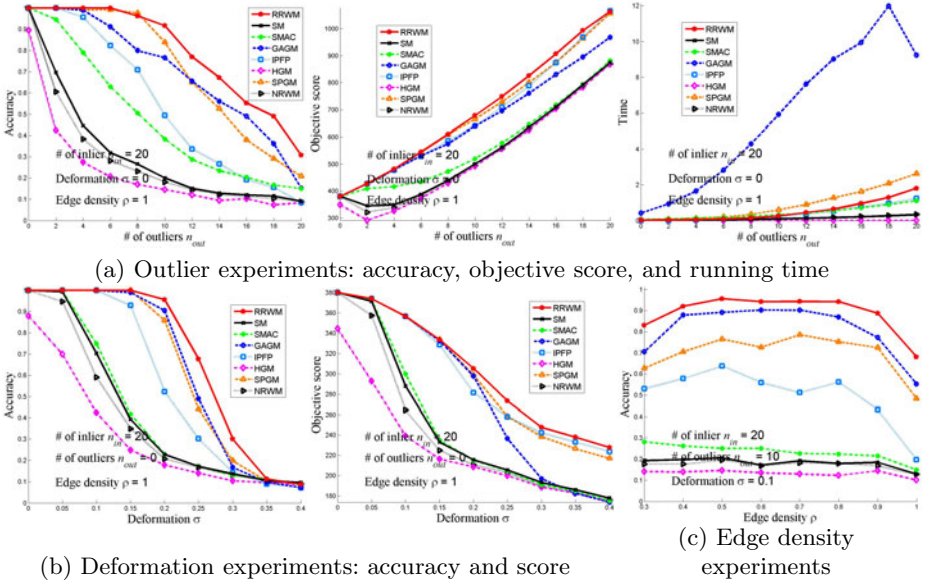
The results are shown in Fig.2. In our experimental setup, there were three kinds of independent variables: outliers  $n_{out}^P$  and  $n_{out}^Q$ , deformation noise  $\sigma$ , and

<sup>3</sup> <http://www.seas.upenn.edu/~timothee/>

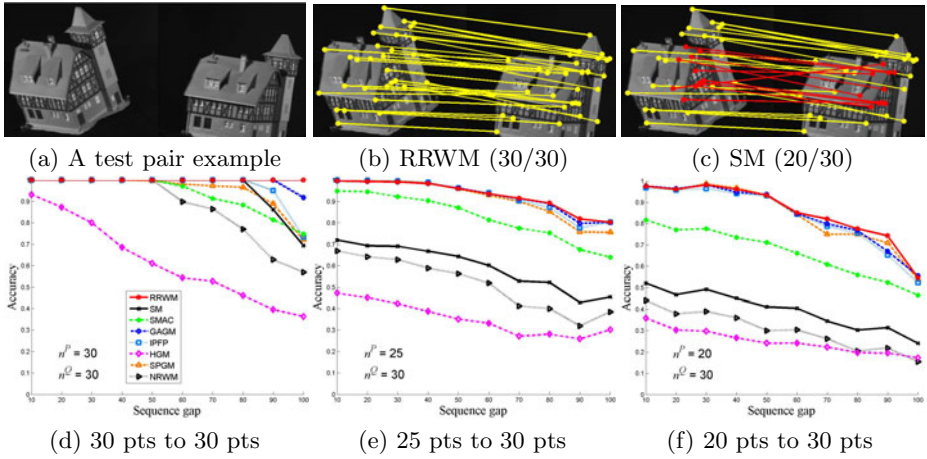
<sup>4</sup> <http://www.cs.huji.ac.il/~zass/>

<sup>5</sup> The original code of HGM [6] uses a high-order affinity, but we edited it to compare with the other methods given the same affinity matrix.

<sup>6</sup> In the comparative experiments in [4], GAGM was commonly used as a post-processing step for the final discretization, but we found that GAGM alone is a strong IQP solver in our experiments. Thus, for observing the performance of each algorithm, we used the Hungarian algorithm [21], the classic linear assignment solver, for the discretization.


**Fig. 2.** Synthetic Graph Matching Experiments

edge density  $\rho$ . Hence, we conducted three sub-experiments to show their influences on performance. For each parameter setting, we generated 100 different matching problems and evaluated the average accuracy and objective score. First, for the outlier experiment in Fig. 2(a), the number of outliers  $n_{out}^P = n_{out}^Q$  were varied from 0 to 20 by increments of 2, while fixing inlier number  $n_{in} = 20$ , deformation noise  $\sigma = 0$ , and edge density  $\rho = 1$ . Second, we experimented on deformation by varying the deformation noise  $\sigma$  from 0 to 0.4 with increments of 0.05 as in Fig. 2(b), while fixing the number of inliers  $n_{in} = 20$ , outliers  $n_{out}^P = n_{out}^Q = 0$ , edge density  $\rho = 1$ . Third, in Fig. 2(c), we varied the edge density  $\rho$  from 0.3 to 1 by increments of 0.1, while the number of inliers  $n_{in} = 20$ , outliers  $n_{out}^P = n_{out}^Q = 10$ , deformation noise  $\sigma = 0.1$ . From all the plots in Fig. 2 with outlier, deformation and edge density variation, we can see that the proposed RRWM outperforms all the other state-of-the-arts methods in both accuracy and objective score. GAGM and SPGM are comparable to RRWM, but with increasing outliers and deformation, RRWM shows consistently better performance. Recent methods based on spectral relaxation and probabilistic interpretation are still less robust to outlier and deformation than GAGM, SPGM, and ours which incorporate the matching constraints in iterative optimizing process. It indicates that tightening relaxation with the matching constraints is an important factor for robust graph matching. As shown in the right plot of Fig. 2(a), RRWM achieves faster convergence than GAGM and SPGM with increasing graph sizes and deformation although all the methods have the same theoretical complexity. We did the same experiments adding outliers only to  $G^Q$  with  $n_{out}^P = 0$ , and the results also show similar trends as in Fig. 2 in all aspects.

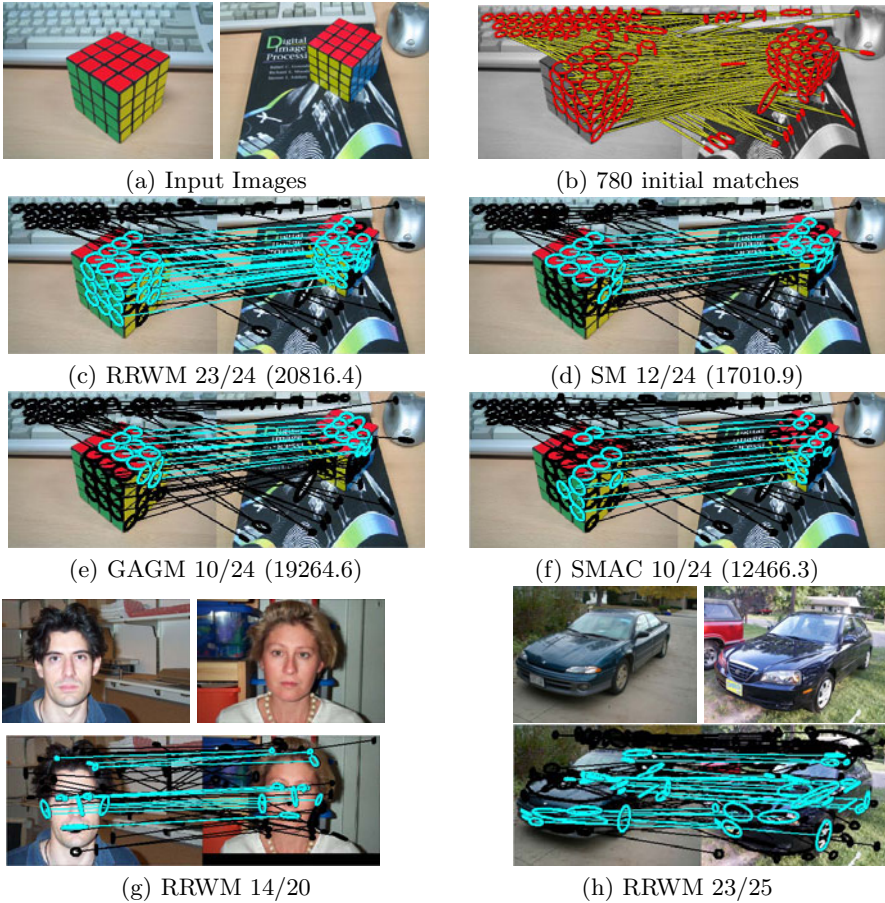


**Fig. 3.** The CMU House sequence experiments

Comparing the performance of SM with NRWM, we can observe the effect of affinity-preserving since SM is equivalent to affinity-preserving random walk matching as proved in Sec.3.1. As shown in all the experiments, affinity-preserving provides more robustness to outlier and deformation than conventional row-normalization. It is consistently demonstrated also in the following experiments.

### 4.2 Feature Point Matching across Image Sequences

In this section, we performed feature point matching on the CMU House sequence which has been widely used in previous works [74] and compared with other methods. In order to assess the matching accuracy, 30 landmark feature points were manually tracked and labeled across all frames. This allows us to compare the performance of the different algorithms over a varying temporal baseline: the larger the temporal baseline between the frames, the larger the relative deformation, and the more difficult the matching. We matched all possible image pairs, total 560 pairs, spaced by 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 frames and computed the average matching accuracy per sequence gap. Graph matching problems for 3 different settings were generated with landmark points as nodes:  $(n^P, n^Q) = (30, 30), (25, 30),$  and  $(20, 30)$ . In the settings for subgraph matching where  $n^P < 30$ , we chose  $n^P$  points randomly among 30 landmark points. The affinity matrix is conducted by  $\mathbf{W}_{ia,jb} = \exp(-|\mathbf{a}_{ij}^P - \mathbf{a}_{ab}^Q|^2 / \sigma_s^2)$ , where  $\mathbf{a}_{ij}^P$  was assigned Euclidean distance between two points. We fixed the scaling factor  $\sigma_s^2 = 2500$  and the edge density  $\rho = 1$ . In this experiment, as  $n^P$  decreases, relative outlier nodes increases. As the sequence gap increases, deformation noise increases. Figure 3 shows the performance curves for  $n^P = 30, 25,$  and  $20$  with respect to the sequence gap. RRWM, GAGM, and SPGM give best performances in this experiment, and RRWM generated perfect matching in the 30 to 30



**Fig. 4.** Some results of real image matching on our dataset. True matches are represented by cyan lines, and false matches by black lines.

problem. Note that as outliers or deformation increases, RRWM, GAGM, SPGM shows larger performance gap from other methods, and RRWM converges faster than both GAGM and SPGM as shown in Fig.2(a).

We also did extensive point matching experiments on random synthetic point sets, and RRWM showed the best performance as similar to this experiments.

### 4.3 Real Image Matching

In this experiment we applied our method to challenging real image matching problems using local feature detectors. We constructed a dataset of 30 image pairs containing various images most of which are collected from Caltech-101 <sup>7</sup>

<sup>7</sup> [http://www.vision.caltech.edu/Image\\_Datasets/Caltech101/](http://www.vision.caltech.edu/Image_Datasets/Caltech101/)

and MSRC<sup>8</sup> datasets, and generated candidate correspondences using the MSER detector [22] and the SIFT descriptor [23]. Using the distance of 128-dim SIFT descriptor, all the possible candidate matches were collected if the feature pair has closer distance in SIFT feature space than a loose threshold  $\delta = 0.6$ , allowing multiple correspondences for each feature.

To measure the dissimilarity between two candidate region correspondences  $(i, a)$  and  $(j, b)$ , we adopted the mutual projection error function  $d_{ia;jb}$  used in [24], and set  $\mathbf{W}_{ia;jb} = \max(50 - d_{ia;jb}, 0)$ . The ground truths were manually labeled for all candidate correspondences of each image pair, and the accuracy and relative objective score were computed and compared with SM, SMAC, and GAGM. The results are summarized in Table I and some representative examples are shown in Fig 4. In Fig 4(c)-(f), the algorithm, true matches per ground truths, and objective scores are captioned. As shown in the examples, this experiment and the dataset are designed for producing the challenging feature matching problems where unary local features are very ambiguous. Our RRWM clearly outperforms other methods both in accuracy and objective score as summarized in Table I. Note that the second best, GAGM was about ten times slower than RRWM in this experiment as similar in the previous experiments.

For the full results of our comparative experiments and more information, refer to our project site: <http://cv.snu.ac.kr/research/~RRWM/>

**Table 1.** Matching performance on the real image dataset (30 pairs)

Methods	RRWM	SM	SMAC	GAGM
Avg. of accuracy (%)	<b>64.01</b>	52.08	39.74	58.74
Avg. of relative score (%)	<b>100</b>	82.41	59.35	91.13

## 5 Conclusion

In this paper, we introduced a graph matching framework based on random walks and proposed a novel graph matching algorithm inspired by the personalized random walks [10,11] and the Sinkhorn method [18]. The experiments demonstrated that it outperforms the state-of-the-art methods [3,4,6,8,13,14] in the presence of outliers and deformation. The comparison reveals that the matching accuracy in the challenging situations largely depends on the effective exploitation of the matching constraints. Our random walk framework is extendable to high-order graph matching adopting the tensor representation as in [7,6]. In our future work, we will improve our framework and method for this direction.

## Acknowledgements

This work was supported by the Korea Research Foundation Grant funded by the Korean Government(MOEHRD) (KRF-2008-314-D00377).

<sup>8</sup> <http://research.microsoft.com/en-us/projects/objectclassrecognition/>

## References

1. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. In: IJPRAI (2004)
2. Berg, A.C., Berg, T.L., Malik, J.: Shape matching and object recognition using low distortion correspondences. In: CVPR (2005)
3. Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. In: ICCV (2005)
4. Cour, T., Srinivasan, P., Shi, J.: Balanced graph matching. In: NIPS (2006)
5. Torresani, L., Kolmogorov, V., Rother, C.: Feature correspondence via graph matching: Models and global optimization. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 596–609. Springer, Heidelberg (2008)
6. Zass, R., Shashua, A.: Probabilistic graph and hypergraph matching. In: CVPR (2008)
7. Duchenne, O., Bach, F., Kweon, I., Ponce, J.: A tensor-based algorithm for high-order graph matching. In: CVPR (2009)
8. Leordeanu, M., Herbert, M.: An integer projected fixed point method for graph matching and map inference. In: NIPS (2009)
9. Kleinberg, J.: Authoritative sources in a hyperlinked environment. *Journal of the ACM* (1999)
10. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical Report, Stanford University (1998)
11. Haveliwala, T.H.: Topic-sensitive pagerank. In: WWW (2002)
12. Maciel, J., Costeira, J.P.: A global solution to sparse correspondence problems. *PAMI* (2003)
13. Gold, S., Rangarajan, A.: A graduated assignment algorithm for graph matching. *PAMI* (1996)
14. van Wyk, B.J., van Wyk, M.A.: A pocs-based graph matching algorithm. *PAMI* (2004)
15. Lee, J., Cho, M., Lee, K.M.: Graph matching algorithm using data-driven markov chain monte carlo sampling. In: ICPR (2010)
16. Gori, M., Maggini, M., Sarti, L.: Exact and approximate graph matching using random walks. *PAMI* (2005)
17. Robles-Kelly, A., Hancock, E.R.: String edit distance, random walks and graph matching. In: IJPRAI (2004)
18. Sinkhorn, R.: A relationship between arbitrary positive matrices and doubly stochastic matrices. *Ann. Math. Statistics* (1964)
19. Langville, A.N., Meyer, C.D.: *Deeper inside pagerank*. Internet Mathematics (2003)
20. Seneta, E.: *Non negative matrices and markov chains*. Springer, Heidelberg (2006)
21. Munkres, J.: *Algorithms for the assignment and transportation problems*. SIAM, Philadelphia (1957)
22. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: BMVC (2002)
23. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV, pp. 1150–1157 (1999)
24. Cho, M., Lee, J., Lee, K.M.: Feature correspondence and deformable object matching via agglomerative correspondence clustering. In: ICCV (2009)

# Rotation Invariant Non-rigid Shape Matching in Cluttered Scenes

Wei Lian<sup>1</sup> and Lei Zhang<sup>2,\*</sup>

<sup>1</sup> Dept. of Computer Science, Changzhi University, Changzhi, 046011, Shanxi, China  
lianwei3@gmail.com

<sup>2</sup> Biometric Research Center, Dept. of Computing, The Hong Kong Polytechnic University, Hong Kong  
cslzhang@comp.polyu.edu.hk

**Abstract.** This paper presents a novel and efficient method for locating deformable shapes in cluttered scenes. The shapes to be detected may undergo arbitrary translational and rotational changes, and they can be non-rigidly deformed, occluded and corrupted by clutters. All these problems make the accurate and robust shape matching very difficult. By using a new shape representation, which involves a powerful feature descriptor, the proposed method can overcome the above difficulties successfully, and it possesses the property of global optimality. The experiments on both synthetic and real data validated that the proposed algorithm is robust to various types of disturbances. It can robustly detect the desired shapes in complex and highly cluttered scenes.

## 1 Introduction

Point matching is a fundamental yet challenging problem in computer vision, pattern recognition and medical image analysis, while non-rigid point matching is particularly difficult due to the large number of possible non-rigid transformations of the template [1]. In this paper, we will address the following problem under the non-rigid point matching framework: locating a deformable shape in cluttered scenes. The shape may undergo arbitrary translational and rotational changes, and it may be non-rigidly deformed, occluded and corrupted by random or structured outliers. All these difficulties make shape matching a formidable task. To overcome these problems, different methods have been proposed [2], which can be classified as those based on local search and those based on global search.

**Methods based on local search.** The iterated closest point (ICP) method [3,4] uses the closest points as the matched points, and it has variants [5,6]. The robust point matching (RPM) method [1] uses deterministic annealing [7] to recover a continuously relaxed point correspondence. The method in [8] uses constraint projection based on quadratic programming to gradually recover the point correspondence and uses clustering for speedup. The covariance driven

---

\* Corresponding author.

correspondence (CDC) method [9] uses the covariance of the transformation parameters to prune the possible false point correspondences. The methods in [10,11] convert point set registration to an image registration problem. These local search methods are generally not rotation invariant and not robust to strong outlier disturbances.

**Methods based on global search.** These methods can be further classified as those based on spatial mapping and those based on point correspondence. For the first category, solution space searching techniques such as genetic algorithm [12], particle filtering [13] and particle swarm optimization [14] can be used to recover the transformation. These methods need no initial coarse alignment and are robust against clutter, but they require an explicit modeling of the transformation and may become computationally expensive when the number of transformation parameters becomes high, which makes them unsuitable for non-rigid matching. The method in [15] constructs a global convex approximation to the matching function and thus the transformations can be optimally recovered. But the number of constraints for the method is usually very high which is circumvented by using interior point methods.

For the second category, linear programming was employed in [16,17] to minimize both the feature matching cost and geometric distortion. Ant colony optimization was employed in [18] for contour correspondence. Dynamic programming (DP) was used to match chain-like or tree-like structures in [19,20]. In [21], it was extended to match regions of a shape. Belief propagation was used in [22] to match shapes where shapes with loops or holes are allowed.

Shape context (SC) [23] is a very informative feature descriptor. The SC of a point is a measure of the distribution of other points relative to it. SC is very discriminative and quite robust to various types of disturbances, which makes it especially useful for non-rigid point matching. However, SC is rotation variant in most applications (i.e. no significant rotations are allowed between two point sets). Attempts at making SC rotation invariant are either susceptible to noise, tend to degrade the discriminative power of SC (e.g. tangent directions were used to determine the orientations of SCs in [23], distance between two SCs was rendered rotation invariant by traversing all rotated versions of one of them and retaining the minimum distance in [17]) or imposing unnatural requirements on point sets (e.g. the directions pointed at the mass center of a point set were used as the orientations of SCs in [24]).

We propose in this paper a new approach to representing shapes and apply it to rotation invariant non-rigid point matching. A shape is triangulated such that the non-boundary edges are long enough and also DP can be used to find the best embedding of the triangles in target point set. Then SC features are constructed for vertices of the triangles whose orientations coincide with the directions of non-boundary edges. The SC features constructed in this way are therefore rotation invariant. To further improve our method's robustness to outliers, we modify the original SC distance measure in [23] such that the SC input belonging to the template is used as a mask to reduce the influence of outliers on the SC input belonging to the target.



Compared with previous attempts at enabling SC rotation invariant, our approach retains the discriminative power of SC, is robust to orientation disturbances and appears natural. It shares similarities with the method in [21] in that both approaches use triangulation to represent shapes and DP is used to find the best embedding of triangles in target set. However, the method in [21] is for deformable template matching in images, and the purpose of triangulation is to introduce non-rigid deformation in template (constrained Delaunay triangulation is adopted to achieve the maximum effect). In comparison, the purpose of triangulation in our method is to render SC rotation invariant, where a different triangulation approach is adopted with the aim that the orientations of SCs should be as robust to disturbances as possible.

The remaining of the paper is organized as follows. Section 2 introduces briefly the shape representation. Section 3 presents a new SC distance measure. Section 4 presents the energy function. Section 5 summarizes the algorithm. Section 6 presents extensive experimental results and section 7 concludes the paper.

## 2 Shape Representation

We restrict ourselves to the cases where the template point set can be represented as a simple polygon, which is a polygon without holes. We call the polygon the boundary of the set. For a general point set, we obtain its boundary by solving the traveling salesman problem [25]. We triangulate the template set such that: 1) its boundary edges are retained; 2) a point is chosen as the reference and the rest points are connected to it (the resulting edges will be called frame edges hereafter). This results in a fan-shaped triangulation. Fig. 1 shows two examples of such triangulation.



**Fig. 1.** Examples of fan-shaped triangulation. The boundaries of the shapes are highlighted in blue, and the frame edges are indicated in black.

We then compute oriented SC [23] for each point except for the reference point, whose positive x-axis is directed at the reference point. Oriented SCs constructed in this way are therefore rotation invariant. Due to the strong discriminative nature of SC, our method’s robustness to various types of disturbances is greatly enhanced.

Alternative ways of triangulation are possible, so why the fan-shaped triangulation is preferred? The orientations of SCs coincide with the directions of frame edges in our method. We know that the longer an edge is, the less likely

its orientation will be affected by positional disturbances of the endpoints. More specifically, assume the endpoints are  $x_i = \hat{x}_i + \Delta x_i, i = 1, 2$ , where  $\hat{x}_i$  denotes the noise free position and  $\Delta x_i$  denotes noise. The direction of the edge is

$$\frac{x_2 - x_1}{\|x_2 - x_1\|} \approx \frac{x_2 - x_1}{\|\hat{x}_2 - \hat{x}_1\|} = \frac{\hat{x}_2 - \hat{x}_1}{\|\hat{x}_2 - \hat{x}_1\|} + \frac{\Delta x_2 - \Delta x_1}{\|\hat{x}_2 - \hat{x}_1\|}$$

The second term comes from noise. Therefore the larger the length  $\|\hat{x}_2 - \hat{x}_1\|$  is, the less influence the noise will impose on the direction of the edge. Fan-shaped triangulation provides a simple and effective solution to ensuring that the edges determining the orientations of SC are long enough. We have also tested several alternative triangulations such as the greedy heuristic based method, where a shape is iteratively divided into two halves by choosing the longest interior edge as the splitting line, but our experimental results demonstrated that fan-shaped triangulation is more robust for point matching.

Based on the same consideration, the reference point in fan-shaped triangulation is chosen such that the average distance from it to the rest points is maximized.

### 3 Outlier Resistant Shape Context Distance

The SC of a point is defined as the distribution of other points relative to it in log-polar coordinate and is quantified as a histogram. Consider two points,  $i$  in template set and  $j$  in target set, their SCs are histograms  $h_i(k)$  and  $h'_j(k)$ , for  $k = 1, 2, \dots, K$ , respectively. The  $\chi^2$  test statistic was used to measure their difference in [23]:

$$\frac{1}{2} \sum_{k=1}^K \frac{[h_i(k) - h'_j(k)]^2}{h_i(k) + h'_j(k)} \tag{1}$$

This measure is effective when there are no outliers or the outliers are homogeneously distributed in target set. But it may become inadequate when there are structured outliers in target set.

To tackle the above problem, based on the observation that the template set is generally outlier free, let us consider the scenario where the only type of disturbance is outliers in target set. If points  $i$  in template set and  $j$  in target set correspond to each other, we would have  $h_i(k) = h'_j(k)$  for all  $k$  if there were no outliers. Since there are outliers in target set, intuitively we can use  $\text{sign}(h_i(k))$  as a mask to reduce the influence of outliers on  $h'_j(k)$ , where

$$\text{sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \end{cases}$$

This is accomplished by replacing  $h'_j(k)$  with  $\hat{h}'_j(k) = \text{sign}(h_i(k)) \cdot h'_j(k)$ . We then normalize  $\hat{h}'_j$  so that it can represent a distribution:  $\sum_{k=1}^K \hat{h}'_j(k) = 1$ . We

now define the outlier resistant shape context distance (ORSCD) between two SCs  $h_i$  and  $h'_j$  as:

$$\frac{1}{2} \sum_{k=1}^K \frac{[h_i(k) - \widehat{h}'_j(k)]^2}{h_i(k) + \widehat{h}'_j(k)} \quad (2)$$

Our experimental results showed that, compared with the original SC distance measure, ORSCD's robustness to outliers is significantly improved while its robustness to non-rigid deformation is only slightly weakened.

## 4 Energy Function

Fan-shaped triangulation will result in a chain of connected triangles, where two triangles are considered connected if they share a common edge, which meets the prerequisite of DP. Therefore DP can be used to find the best embedding of these triangles in target point set. In this section, we present the energy function associated with the matching problem.

Suppose that the 2D template point set is  $\mathcal{X} = \{x_i, 0 \leq i \leq n\}$ , where the sequence  $x_0, x_1, \dots, x_n, x_0$  forms its closed boundary. Without loss of generality,  $x_0$  is assumed to be the reference. Denote by  $\mathcal{Y} = \{y_j, 0 \leq j \leq m\}$  the point set to be matched. The task of matching is to find a mapping  $\phi: \mathcal{X} \rightarrow \mathcal{Y}$  which maps the  $i$ th point in  $\mathcal{X}$  to the  $l_i$ th point in  $\mathcal{Y}$  so that certain energy function can be minimized.

The energy function used in our method is

$$E(\phi) = E_{sc}(\phi) + \lambda E_{bound}(\phi) + \mu E_{frame}(\phi) \quad (3)$$

where the term  $E_{sc}$  penalizes the SC distance between the matched points, the term  $E_{bound}$  and  $E_{frame}$  require, respectively, that the lengths of boundary and frame edges should be preserved during matching. The constants  $\lambda$  and  $\mu$  ( $\lambda \geq 0, \mu \geq 0$ ) serve to balance the weights of the three terms. (We assume that the template point set is unit sized and choose  $\lambda = 1, \mu = 0.5$  in our method). For non-rigid matching, a smaller  $\mu$  allows for more non-rigid behavior of the method.

The term  $E_{sc}$  is defined as:

$$E_{sc}(\phi) = \sum_{i=1}^n D_{sc}[i, 0](l_i, l_0) \quad (4)$$

where  $D_{sc}[i, 0](l_i, l_0)$  denotes the original SC distance [23] or ORSCD between the oriented SC of  $x_i$  and the oriented SC of  $y_{l_i}$ . The positive x-axis of SC for  $x_i$  is directed at  $x_0$ , and the positive x-axis of SC for  $y_{l_i}$  is directed at  $y_{l_0}$ . The SC distances computed in this way are therefore rotation invariant.

The term  $E_{bound}$  is defined as:

$$E_{bound}(\phi) = \sum_{i=0}^{n-1} D_{bound}[i, i+1](l_i, l_{i+1}) \quad (5)$$

where  $D_{bound}[i, i+1](l_i, l_{i+1})$  denotes the length difference between the boundary edge  $(i, i+1)$  in  $\mathcal{X}$  and the candidate edge  $(l_i, l_{i+1})$  in  $\mathcal{Y}$ :

$$D_{bound}[i, i+1](l_i, l_{i+1}) = \left| \|y_{i+1} - y_i\| - \|x_{i+1} - x_i\| \right| \quad (6)$$

If the length of a boundary edge  $(i, i+1)$  in  $\mathcal{X}$  is close to 0, which often occurs in contour matching,  $D_{bound}$  can be further simplified as:

$$D_{bound}[i, i+1](l_i, l_{i+1}) = \|y_{i+1} - y_i\| \quad (7)$$

The term  $E_{frame}$  is defined as:

$$E_{frame}(\phi) = \sum_{i=2}^n D_{frame}[i, 0](l_i, l_0) \quad (8)$$

where  $D_{frame}[i, 0](l_i, l_0)$  denotes the length difference between the frame edge  $(i, 0)$  in  $\mathcal{X}$  and the candidate edge  $(l_i, l_0)$  in  $\mathcal{Y}$ . We use the  $\chi^2$  test statistic [23] instead of the Euclidean distance to measure the length difference:

$$D_{frame}[i, 0](l_i, l_0) = \frac{\left| \|y_i - y_{l_0}\| - \|x_i - x_0\| \right|^2}{\|y_i - y_{l_0}\| + \|x_i - x_0\|} \quad (9)$$

This is based on the fact that shorter edges are less distorted than longer edges under a non-rigid deformation. Therefore the length differences of shorter edges should be penalized more than those of longer edges.

## 5 Algorithm

During initialization, oriented SC is constructed for each point in  $\mathcal{X}$  with  $x_0$  serving as the reference, which has time complexity  $O(n)$  and space complexity  $O(n)$ . Oriented SC is then constructed for each point in  $\mathcal{Y}$  with all the rest points serving as possible references, which has time complexity  $O(m^2)$  and space complexity  $O(m^2)$ . Finally, distances between oriented SC features in both point sets are computed, which has time complexity  $O(nm^2)$  and space complexity  $O(nm^2)$ .

In practice, the time of computing SC features for a point in  $\mathcal{Y}$  with all the rest points serving as possible references can be reduced by quantizing orientation into  $M$  evenly distributed angles ( $M = 50$  is chosen in our method):  $0, \frac{1}{M}2\pi, \dots, \frac{M-1}{M}2\pi$ , and only computing SC features with these angles as the possible orientations. Then the SC features with all the rest points being possible references are substituted by these SC features based on orientation proximity. With this heuristic, the complexity of the initialization is essentially  $O(nm)$ .

SC distances are then used in the optimization. The algorithm is an instantiation of the well known DP technique. We compute the cost of the best placements  $l_j$  for  $j = 1, \dots, i-1$  as a function of the placements  $l_0$  and  $l_i$ , which is stored in  $V[i, 0](l_i, l_0)$ . The algorithm is summarized as follows.

**Algorithm 1.** Find the best embedding of a shape in a point set

1.  $V[1, 0](l_1, l_0) = D_{sc}[1, 0](l_1, l_0) + \lambda D_{bound}[0, 1](l_0, l_1)$
2. For  $i = 2, \dots, n$ , do
  - $V[i, 0](l_i, l_0) \leftarrow \min_{l_{i-1}} V[i-1, 0](l_{i-1}, l_0) + \lambda D_{bound}[i-1, i](l_{i-1}, l_i)$  ;
  - $V[i, 0](l_i, l_0) \leftarrow V[i, 0](l_i, l_0) + D_{sc}[i, 0](l_i, l_0) + \mu D_{frame}[i, 0](l_i, l_0)$
3. Pick  $l_n$  and  $l_0$  minimizing  $V[n, 0]$  and trace back to obtain the other optimal locations.

The above procedure has time complexity  $O(nm^3)$  and space complexity  $O(nm^2)$ . We can speed it up based on two considerations: First, if the length of a boundary edge  $(i-1, i)$  in  $\mathcal{X}$  is close to 0, given location  $l_i$ , the possible candidates for  $l_{i-1}$  should be those points near  $y_i$  [21], because points that are far from it will introduce too much distortion in the template (15 nearest points are chosen in our method). Second, given location  $l_i$ , the possible candidates for  $l_0$  should be those points which are close to the circle centered at  $y_i$  and with a radius equal to the length of the edge  $(i, 0)$  in  $\mathcal{X}$ , because points that are far from the circle will also introduce too much distortion in the template. With the two heuristics, the complexity of the proposed algorithm is essentially  $O(nm)$ .

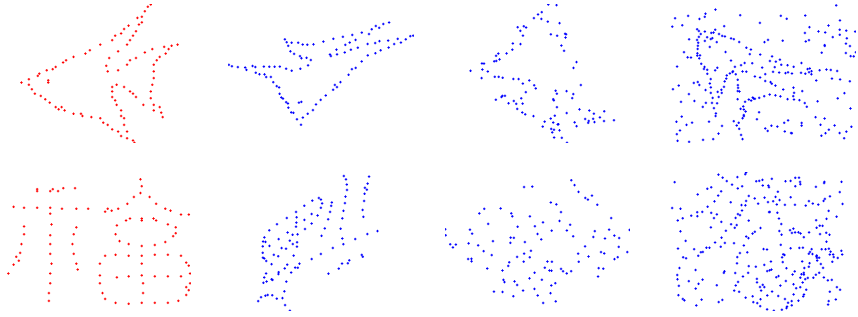
## 6 Experimental Results

We compare our method with 3 state-of-the-art methods: the local neighborhood structure preserving (LNSP) method in [24], the Viterbi algorithm (VA) based method in [26], and the linear programming (LP) based method in [16] where we choose SC as the feature descriptor. VA and LP are not rotation invariant. We render them rotation invariant by running them on 12 evenly distributed angles and retaining the result with the minimum cost. The code of our method is available at <http://www4.comp.polyu.edu.hk/~cs1zhang/code.htm>

We implement the methods under Matlab version 7.6 on a PC with 2GHz CPU and 2G memory. We use affine transformation to model a non-rigid spatial mapping. Correspondence recovered by a method is used to solve for the affine transformation. In the following, the transformed template point set is highlighted by red \* and point correspondences are indicated by black line segments. First we use synthetic data to evaluate various aspects of the methods. Then we compare the methods using data acquired from real images.

### 6.1 Experiments Using Synthetic Data

Synthetic data can be designed to test specific aspects of a method. First, we use the Chui-Rangarajan synthesized data sets [1] to test the methods' robustness against non-rigid **deformation**, **noise** in position and **outliers**. In each test, the template shape is subjected to one of the above distortions to create a target point set (for the latter two test sets, a moderate amount of deformation is present). Two shapes, a fish and a Chinese character, shown in the left column



**Fig. 2.** The template point sets (left column) and examples of target point sets in the deformation, noise and outlier tests respectively (right 3 columns)

of Fig. 2 are used as the template shape respectively. 100 random target point sets were generated for each setting within each series. The right 3 columns of Fig. 2 show examples of target point sets in the 3 series of tests respectively. We use the original SC distance measure in our method.

The means and standard deviations of the errors of the methods are shown in Fig. 3, where error is defined as the mean of the Euclidean distances between the affinely transformed template points and their ground truth target points. It can be seen that the matching error of our method is in average compared with the other methods for the deformation and noise tests, while considerably lower than others for the outlier test. This demonstrates our method's robustness against various types of disturbances, especially for outliers.

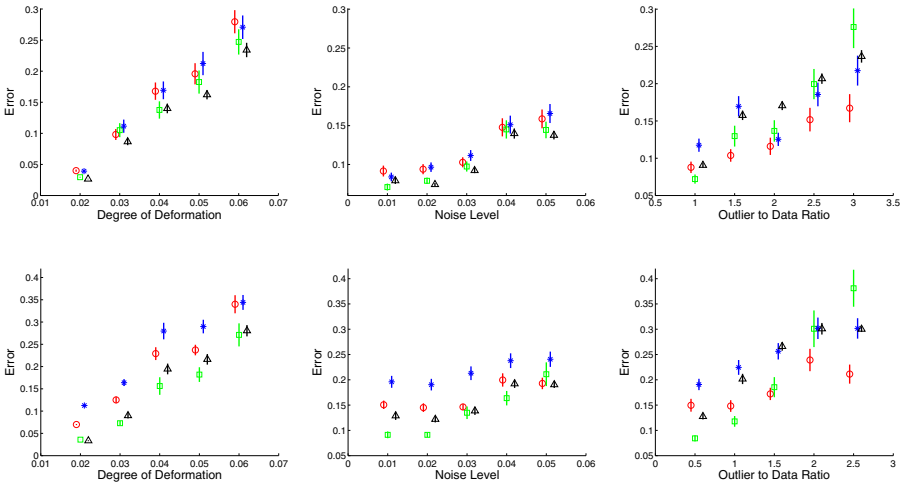
The average running times of the methods are listed in Table 1. It can be seen that our method's running time is low when the number of points is low, but increases much when the number of points becomes high (i.e. in the case of outliers).

**Table 1.** Average Running Time (second)

	Deformation	Noise	Outliers
LNSP	4.0622	5.1435	28.0950
LP	35.1288	35.4172	67.1175
VA	6.9798	7.0181	25.2389
Our method	7.1719	7.0601	36.5388

We then test the methods' robustness against complex clutters. Two shapes similar to the template shape but with different poses (the most similar one is indicated in red in the figure) are mixed together to generate the target point sets. Random outliers are then added to the target point sets. The aim is to

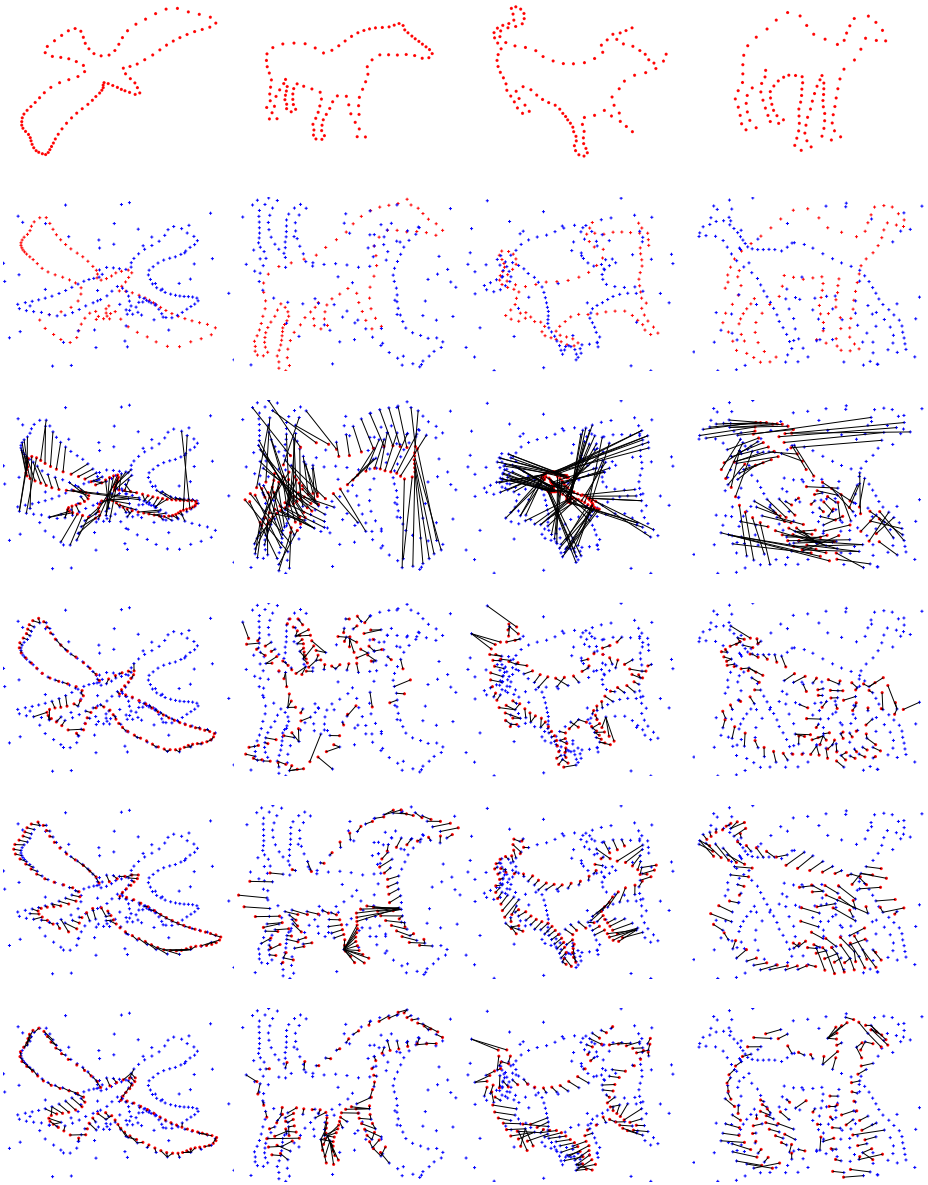
animate complex clutter. We use the original SC distance measure in our method. Examples of shape matching by all the methods are shown in Fig. 4. It can be seen that, in addition to non-rigid deformation and random outliers, the mixing of similar shapes considerably complicates the matching problem. Despite the difficulties, our method works much better at matching the template shapes to the correct target shapes than the other methods, validating the robustness of our method against complex clutters.



**Fig. 3.** Comparison of our method (red  $\circ$ ) with VA (green  $\square$ ), LP (blue  $*$ ) and LNSP (black  $\triangle$ ) on the Chui-Rangarajan synthesized data sets. The error bars indicate the standard deviation of the error over 100 random trials. Top row: fish tests. Bottom row: Chinese character tests.

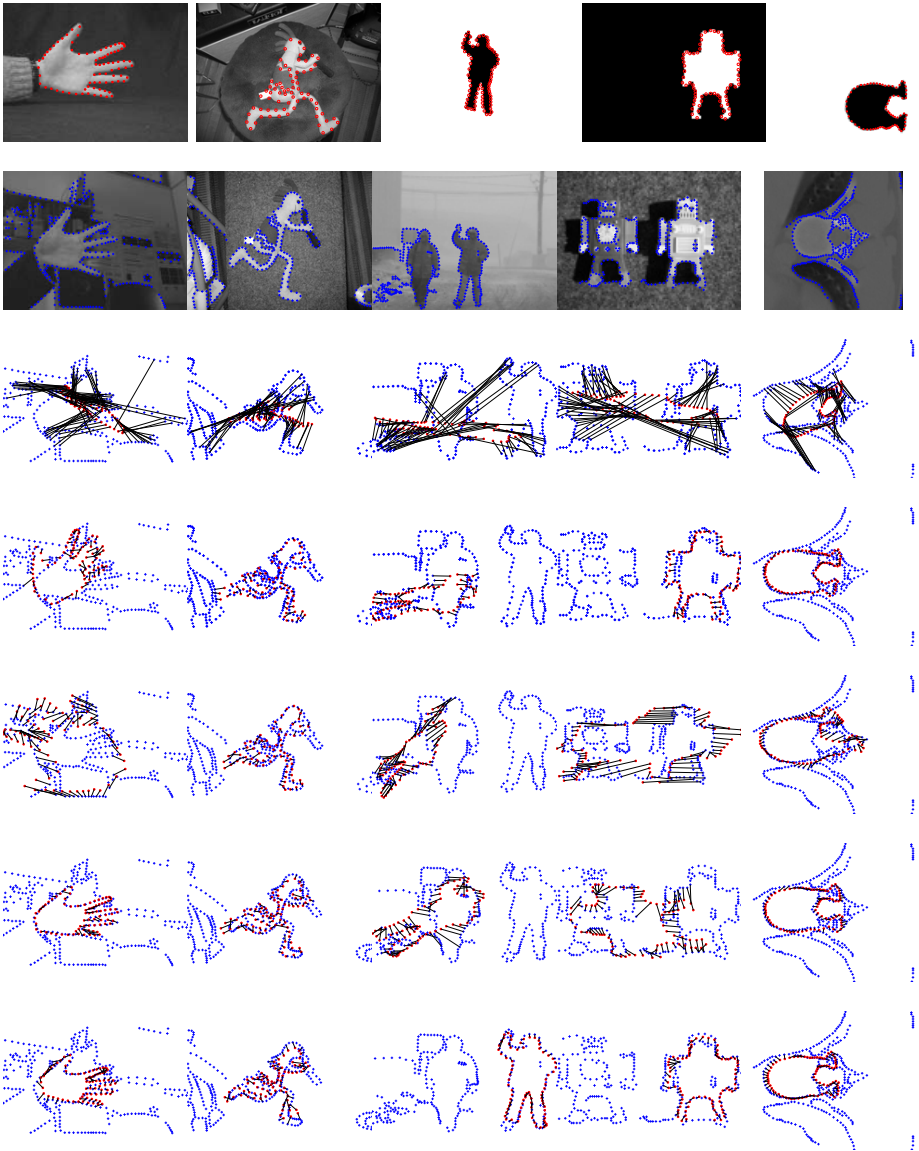
### 6.2 Experiments on Real Data

We finally test the methods’ performance using data acquired from images. Examples of matching results by the competing methods are shown in Fig. 5, where the template shapes are further randomly rotated (not shown in the figure) with the aim at testing the methods’ abilities for solving rotations. It can be seen that our method using ORSCD can successfully match to the correct shapes for all the tests, while our method using the original SC distance measure fails for the 3rd and 4th tests where similar shapes coexist in the same picture. This demonstrates ORSCD’s robustness against structured outliers compared with the original SC distance measure. In comparison, LP fails for the 1st and 3rd tests, VA only succeeds for the 2nd test, and LNSP fails for all the tests. This clearly shows our method’s potential for rotation invariant non-rigid shape matching arising from real problems.



**Fig. 4.** Examples of point matching in case of complex clutter. The first row shows the template shapes. The second row shows the mixture of two shapes which are similar to the template shapes (the most similar ones are indicated in red) and random outliers. The last 4 rows show the matching results by LNSP, LP, VA and our method respectively.





**Fig. 5.** Examples of point matching with data acquired from images. The first row shows the images used to extract the template point sets (red \*). The second row shows the images used to extract the target point sets (blue +). Points are extracted via Canny edge detector. The last 5 rows show the matching results by LNSP, LP, VA and our method using the original SC distance measure and ORSCD respectively.

## 7 Conclusion

We proposed a novel and efficient method for representing and matching non-rigid shapes. The representation is invariant to translational and rotational changes, and by using a powerful feature descriptor and a new feature distance measure, it is also robust to non-rigid deformations and outliers. An algorithm was then proposed to solve the point matching problem, which possesses global optimality and is very robust against clutters. The proposed method was tested by using both simulated and real data in comparison with 3 state-of-the-art and representative methods. The results clearly demonstrated that the proposed method has high capability in detecting and matching shapes in cluttered scenes.

In the future, we will apply the proposed method to matching other types of rotation variant features such as local image patch and geometric blur.

## Acknowledgements

This research is supported by the Hong Kong RGC General Research Fund (PolyU 5351/08E) and the Hong Kong Polytechnic University Internal Fund (A-SA08).

## References

1. Chui, H., Rangarajan, A.: A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding* 89, 114–141 (2003)
2. Veltkamp, R.C., Hagedoorn, M.: State of the art in shape matching, pp. 87–119 (2001)
3. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence* 14, 239–256 (1992)
4. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision* 13, 119–152 (1994)
5. Stewart, C.V., Tsai, C.L., Roysam, B.: The dual-bootstrap iterative closest point algorithm with application to retinal image registration. *IEEE Trans. Medical Imaging* 22, 1379–1394 (2003)
6. Fitzgibbon, A.W.: Robust registration of 2d and 3d point sets. *Image and Vision Computing* 21, 1145–1153 (2003); *British Machine Vision Computing* 2001 (2001)
7. Yuille, A.L., Kosowsky, J.J.: Statistical physics algorithms that converge. *Neural Comput.* 6, 341–356 (1994)
8. Lian, W., Zhang, L., Liang, Y., Pan, Q.: A quadratic programming based cluster correspondence projection algorithm for fast point matching. *Computer Vision and Image Understanding* 114, 322–333 (2010)
9. Sofka, M., Yang, G., Stewart, C.V.: Simultaneous covariance driven correspondence (cdc) and transformation estimation in the expectation maximization framework. In: *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8 (2007)
10. Tsin, Y., Kanade, T.: A correlation-based approach to robust point set registration. In: Pajdla, T., Matas, J(G.) (eds.) *ECCV 2004*. LNCS, vol. 3023, pp. 558–569. Springer, Heidelberg (2004)
11. Jian, B., Vemuri, B.C.: A robust algorithm for point set registration using mixture of gaussians. In: *IEEE International Conference on Computer Vision*, vol. 2, pp. 1246–1251 (2005)

12. Silva, L., Bellon, O.R., Boyer, K.L.: Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence* 27, 762–776 (2005)
13. Sandhu, R., Dambreville, S., Tannenbaum, A.: Point set registration via particle filtering and stochastic dynamics. *IEEE Trans. Pattern Analysis and Machine Intelligence* 32, 1459–1473 (2010)
14. Li, H., Shen, T., Huang, X.: Global optimization for alignment of generalized shapes. In: *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 856–863 (2009)
15. Taylor, C.J., Bhusnurmath, A.: Solving image registration problems using interior point methods. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part IV*. LNCS, vol. 5305, pp. 638–651. Springer, Heidelberg (2008)
16. Jiang, H., Drew, M.S., Li, Z.N.: Matching by linear programming and successive convexification. *IEEE Trans. Pattern Analysis and Machine Intelligence* 29, 959–975 (2007)
17. Jiang, H., Yu, S.X.: Linear solution to scale and rotation invariant object matching. In: *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 2474–2481 (2009)
18. Kaick, O.v., Hamarneh, G., Zhang, H., Wightton, P.: Contour correspondence via ant colony optimization. In: *PG 2007: Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, pp. 271–280 (2007)
19. Scott, C., Nowak, R.D.: Robust contour matching via the order-preserving assignment problem. *IEEE Trans. Image Processing* 15, 1831–1838 (2006)
20. Wang, J., Athitsos, V., Sclaroff, S., Betke, M.: Detecting objects of variable shape structure with hidden state shape models. *IEEE Trans. Pattern Analysis and Machine Intelligence* 30, 477–492 (2008)
21. Felzenszwalb, P.F.: Representation and detection of deformable shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence* 27, 208–220 (2005)
22. Coughlan, J.M., Ferreira, S.J.: Finding deformable shapes using loopy belief propagation. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*. LNCS, vol. 2352, pp. 453–468. Springer, Heidelberg (2002)
23. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence* 24, 509–522 (2002)
24. Zheng, Y., Doermann, D.: Robust point matching for nonrigid shapes by preserving local neighborhood structures. *IEEE Trans. Pattern Analysis and Machine Intelligence* 28, 643–649 (2006)
25. [http://en.wikipedia.org/wiki/Traveling\\_salesman\\_problem](http://en.wikipedia.org/wiki/Traveling_salesman_problem)
26. Thayananthan, A., Stenger, B., Torr, P.H.S., Cipolla, R.: Shape context and chamfer matching in cluttered scenes. In: *IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 127–133 (2003)

# Loosely Distinctive Features for Robust Surface Alignment

Andrea Albarelli, Emanuele Rodolà, and Andrea Torsello

Dipartimento di Informatica - Università Ca' Foscari di Venezia

**Abstract.** Many successful feature detectors and descriptors exist for 2D intensity images. However, obtaining the same effectiveness in the domain of 3D objects has proven to be a more elusive goal. In fact, the smoothness often found in surfaces and the lack of texture information on the range images produced by conventional 3D scanners hinder both the localization of interesting points and the distinctiveness of their characterization in terms of descriptors. To overcome these limitations several approaches have been suggested, ranging from the simple enlargement of the area over which the descriptors are computed to the reliance on external texture information. In this paper we offer a change in perspective, where a game-theoretic matching technique that exploits global geometric consistency allows to obtain an extremely robust surface registration even when coupled with simple surface features exhibiting very low distinctiveness. In order to assess the performance of the whole approach we compare it with state-of-the-art alignment pipelines. Furthermore, we show that using the novel feature points with well-known alternative non-global matching techniques leads to poorer results.

## 1 Introduction

Feature detection and characterization is a key step in many tasks involving the recognition, registration or database search of 2D and 3D data. Specifically, when suitable interest points are available, all these problems can be tackled by working with the set of extracted features, rather than dealing with the less stable and noisier information carried by the whole data. Of course, for an interest point to be reliable it must exhibit two properties: repeatability and distinctiveness. A feature is highly repeatable if it can be detected with good positional accuracy over a wide range of noise levels and sampling conditions as well as different scales and transformations of the data itself. Further, description vectors calculated over interesting points are said to be distinctive if they are well apart when related to different features, yet coherent when associated to multiple instances of the same point. These properties are somewhat difficult to attain since they are subject to antithetical goals. In fact, to achieve good repeatability despite of noise, larger patches of data must be considered. Unfortunately this leads to a lower positional precision and a less sharp culling of uninteresting points. Moreover, for descriptor vectors to be distinctive among different features, they

need to adopt a large enough basis, which, owing to the well known “dimensionality curse”, also affects their coherence over perturbed versions of the same feature. In the last two decades these quandaries have been addressed with great success in the domain of 2D images where salient points are localized with sub-pixel accuracy by detectors exploiting strong local variation in intensity, such as Harris Operator [1] and Difference of Gaussians [2], or by using techniques that are able to locate affine invariant regions, such as Maximally stable extremal regions (MSER) [3] and Hessian-Affine [4]. Among the most used descriptors are the Scale-invariant feature transform (SIFT) [5], the Speeded Up Robust Features (SURF) [6] and Gradient Location and Orientation Histogram (GLOH) [7]. While these approaches work well with 2D intensity images, they cannot be easily extended to handle 3D surfaces since no intensity information is directly available. Of course several efforts have been made to use other local measures, such as curvature or normals. One of the first descriptor to capture the structural neighborhood of a surface point was described by Chua and Jarvis that with their Point Signatures [8] suggest both a rotation and translation invariant descriptor and a matching technique. Later, Johnson and Hebert introduced Spin Images [9], a rich characterization obtained by a binning of the radial and planar distances of the surface samples respectively from the feature point and from the plane fitting its neighborhood. Given their ability to perform well with both surface registration and object recognition, Spin Images have become one of the most used 3D descriptors. More recently, Pottmann et al. proposed the use of Integral Invariants [10], stable multi-scale geometric measures related to the curvature of the surface and the properties of its intersection with spheres centered on the feature point. Finally, Zaharescu et al. [11] presented a comprehensive approach for interest point detection (MeshDOG) and description (MeshHOG), based on the value of any scalar function defined over the surface (i.e. curvature or texture, if available). MeshDOG localizes feature points by searching for scale-space extrema over progressive Gaussian convolutions of the scalar function and thus by applying proper thresholding and corner detection. MeshHOG calculates a histogram descriptor by binning gradient vectors with respect to a rotational invariant local coordinate system.

In this paper we introduce a novel technique to detect and describe 3D interest points and to use them for robust surface registration. Unlike previous approaches we do not aim to obtain a very distinctive characterization. Instead, we settle for very simple descriptors, named *Surface Hashes*, that span only 3 to 5 dimensions. As their name suggests, we expect Surface Hashes to be repeatable through the same feature point, yet to suffer a high level of clashing due to their limited distinctiveness. In order to overcome this liability we avoid the use of classical RANSAC-based matchers; rather we adopt a robust game-theoretic inlier selector which exploits rigidity constraints among surfaces to guarantee a global geometric consistency. The combination of these loosely distinctive features and our robust matcher leads to an effective surface alignment approach. In the experimental section we point out this symbiosis by showing that standard matching techniques are not able to make the most of our descriptors.

## 2 Game-Theoretic Matching

Before describing in detail the Surface Hashes features we need to introduce some basic concepts about Evolutionary Game Theory and to present the idea of a Matching Game, originally presented in [12] and exploited by our technique both as an inlier selector and a robust matcher.

Evolutionary Game Theory [13] considers an idealized scenario where pairs of individuals are repeatedly drawn at random from a large population to play a two-player game. Each player obtains a payoff that depends only on the strategies played by him and its opponent. Players are not supposed to behave rationally, but rather they act according to a pre-programmed behavior, or mixed strategy. It is supposed that some selection process operates over time on the distribution of behaviors favoring players that receive larger payoffs. More formally, let  $S = \{1, \dots, n\}$  be the set of available strategies (*pure strategies* in the language of game theory) and  $C = (c_{ij})$  be a matrix specifying the payoff that an individual playing strategy  $i$  receives against someone playing strategy  $j$ . A *mixed strategy* is a probability distribution  $\mathbf{x} = (x_1, \dots, x_n)^T$  over the available strategies  $S$ ; being probability distributions, mixed strategies lie in the  $n$ -dimensional standard simplex  $\Delta^n = \{\mathbf{x} \in \mathbb{R}^n : \forall i \in 1 \dots n \ x_i \geq 0, \sum_{i=1}^n x_i = 1\}$ . The *support* of a mixed strategy  $\mathbf{x} \in \Delta$ , denoted by  $\sigma(\mathbf{x})$ , is defined as the set of elements chosen with non-zero probability:  $\sigma(\mathbf{x}) = \{i \in S \mid x_i > 0\}$ . The expected payoff received by a player choosing element  $i$  when playing against a player adopting a mixed strategy  $\mathbf{x}$  is  $(C\mathbf{x})_i = \sum_j c_{ij}x_j$ , hence the expected payoff received by adopting the mixed strategy  $\mathbf{y}$  against  $\mathbf{x}$  is  $\mathbf{y}^T C\mathbf{x}$ . The *best replies* against mixed strategy  $\mathbf{x}$  is the set of mixed strategies

$$\beta(\mathbf{x}) = \{\mathbf{y} \in \Delta \mid \mathbf{y}^T C\mathbf{x} = \max_{\mathbf{z}} (\mathbf{z}^T C\mathbf{x})\}.$$

A strategy  $\mathbf{x}$  is said to be a *Nash equilibrium* if it is the best reply to itself, i.e.,  $\forall \mathbf{y} \in \Delta, \mathbf{x}^T C\mathbf{x} \geq \mathbf{y}^T C\mathbf{x}$ . This implies that  $\forall i \in \sigma(\mathbf{x})$  we have  $(C\mathbf{x})_i = \mathbf{x}^T C\mathbf{x}$  that is, the payoff of every strategy in the support of  $\mathbf{x}$  is constant. A strategy  $\mathbf{x}$  is said to be an *evolutionary stable strategy* (ESS) if it is a Nash equilibrium and  $\forall \mathbf{y} \in \Delta \quad \mathbf{x}^T C\mathbf{x} = \mathbf{y}^T C\mathbf{x} \Rightarrow \mathbf{x}^T C\mathbf{y} > \mathbf{y}^T C\mathbf{y}$ . This condition guarantees that any deviation from the stable strategies does not pay. The search for a stable state is performed by simulating the evolution of a natural selection process. Under very loose conditions, any dynamics that respect the payoffs is guaranteed to converge to Nash equilibria [13] and (hopefully) to ESS's; for this reason, the choice of an actual selection process is not crucial and can be driven mostly by considerations of efficiency and simplicity. We chose to use the replicator dynamics, a well-known formalization of the selection process governed by the following equation

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) \frac{(C\mathbf{x}(t))_i}{\mathbf{x}(t)^T C\mathbf{x}(t)}$$

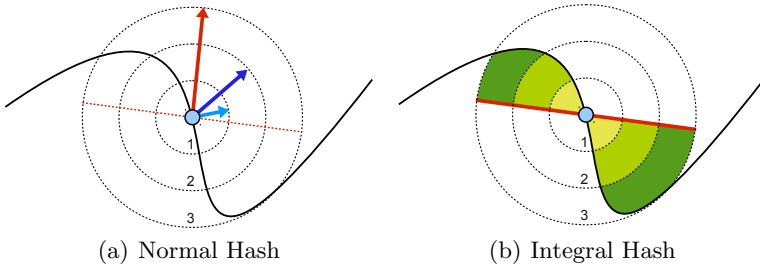
where  $\mathbf{x}_i$  is the  $i$ -th element of the population and  $C$  the payoff matrix.

Once the population has reached a local maximum, all the non-extinct pure strategies (i.e.  $\sigma(\mathbf{x})$ ) can be considered selected by the process.

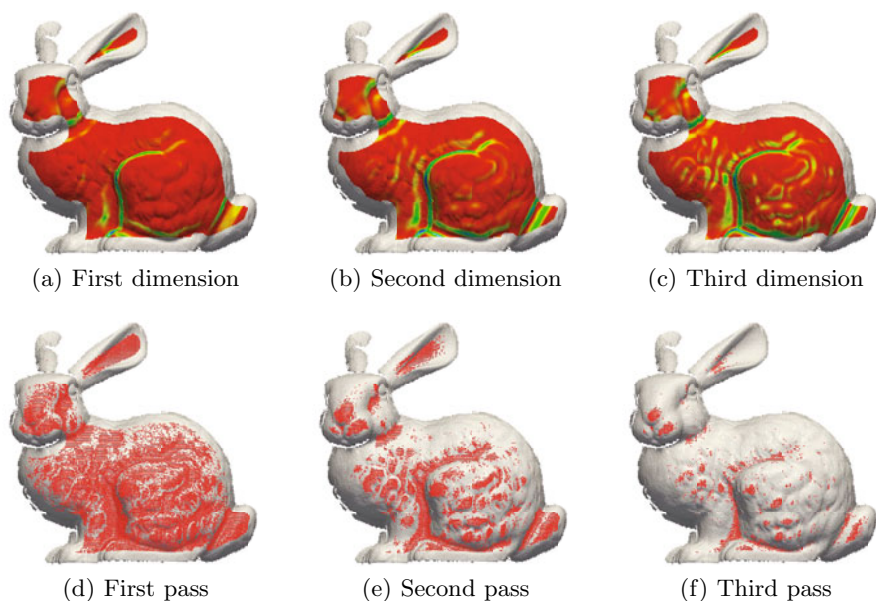
Following [12] and [14], we define a *Matching Game* as a non-cooperative game where the set of strategies  $S$  is a subset of all the possible correspondences, and the payoff  $c_{ij}$  between two strategies is proportional to some notion of compatibility between correspondences. By using different sets to be matched and alternative payoff functions, we are able to define games specially crafted to solve specific problems. In the following section we will define more formally two Matching Games. Respectively the first game will be dedicated to the localization of interest points over a surface described by Surface Hashes, while the second one will address the search for reliable correspondences between feature points extracted from two different meshes.

### 3 Surface Hashes

Intuitively, a Surface Hash is a concise point feature descriptor which exhibit the property of being highly repeatable at the cost of a relatively high probability of clashing. In practice this happens with any low-dimensional descriptor, such as the Gaussian or Mean Curvature (1 dimension), the first two Principal Components of a patch (2 dimensions), or the normal vector associated to a point (2 dimensions). While those descriptors could be used with our registration pipeline, we prefer to introduce some multiscale Surface Hashes based respectively on the dot product between normals and a local surface integral. Each of our descriptors corresponds to a vector of scalar measures evaluated at different scales. By increasing or reducing the number of scales, we are able to obtain vectors of different length, thus being more or less distinctive. The *Normal Hash* (see Fig. 1(a)) is obtained by setting as a reference the average surface normal over a patch that extends to the largest scale (red arrow in figure) and then, for each smaller scale, calculate the dot product between the reference and the average normal over the reduced patches (blue arrows in figure). This measure finds its rationale in the observation that at the largest scale the average normal is more stable with respect to noise and that the dot product offers a concise representation of the relation between the vectors obtained at various scales. The *Integral Hash* (see Fig. 1(b)) is similar in spirit to the Normal Hash. In this case we search for the best fitting plane (in the least squares sense) with respect



**Fig. 1.** Example of the two basic Surface Hashes proposed in this paper



**Fig. 2.** Example of a 3-dimensional Normal Hash and the related detection process

to the surface patch associated to the largest scale. We calculate the volume enclosed between the surface and such a plane. In practice, it is not necessary to evaluate this volume accurately: even naive approximations, such as the sum of the distances of the surface points from the plane, have shown to provide a reasonable approximation in all the empirical tests. Note that Normal Hashes evaluated over  $n$  scales yield descriptor vectors of length  $n - 1$  (since the larger scale is used only to calculate the reference normal), while Integral Hashes provide  $n$ -dimensional vectors. In Fig. 2 a Normal Hash of dimension 3 (respectively from (a) to (c)) evaluated over 4 scales is shown. Note that the descriptor is not defined on the points for which the larger support is not fully contained in the surface, i.e., points close to the surface boundary.

### 3.1 Interest Points Detection

Given the large number of points contained in typical 3D objects, it is not practical for any matching algorithm to deal with all of them. In addition, the isolation of a relatively small number of interest points can enhance dramatically the ability of the matcher to avoid false correspondences, usually due to a large number of features with very common characterizations. This is particularly true when using Surface Hashes, which are loosely distinctive by design. Paradoxically, we use exactly this property to screen out features exhibiting descriptors that are too common over the surface. This happens by defining a Matching Game



where the strategy set  $S$  corresponds to the set of all the surface points and the payoff matrix is defined by:

$$C(ij) = e^{-\alpha|d_i-d_j|}, \quad (1)$$

where  $d_i$  and  $d_j$  are the descriptor vectors associated to surface point  $i$  and  $j$ , and  $\alpha$  is a parameter that controls the level of selectivity. Clearly, features that are similar in terms of Surface Hashes will get a large mutual payoff and thus are more likely to be selected by the evolutive process. In this sense, our goal is to let the population evolve to an ESS and then remove from the set of interest points the features that survived the evolutive process. At the beginning we can initialize the set of retained features to the whole surface and run a sequence of Matching Games until the desired number of points are left. At this point, the remaining features are those characterized by less-common descriptors which are more likely to represent good cues for the matching. It should be noted that by choosing high values for  $\alpha$  the payoff function decreases more rapidly with the growth of the distance between the Surface Hashes, thus the Matching Game becomes more selective and less points survive after reaching an ESS. In the end this results in a blander decimation and thus in a larger ratio of retained interest points. By converse, a low value for  $\alpha$  leads to a more greedy filtering and thus to a more selective interest point detector. In Fig. 2 (from (d) to (f)) we show three steps of the evolutive interest point selection with respect to the 3-dimensional Normal Hash shown from (a) to (c). In Fig. 2(d) we see that after a single pass of the Matching Game most of the surface points are still considered interesting, while after respectively two and three passes only very distinctive points (belonging to areas with less common curvatures) are left.

### 3.2 Matching Surface Hashes

After obtaining a reduced set of interest points from the two surfaces, we could proceed to align them using some robust algorithm such as a basic RANSAC [15], that would use just the point locations and some initial match hypotheses, or PROSAC [16], that could better exploit the prior expressed by the descriptors. Unfortunately, Surface Hashes, despite the proposed filtering technique, are still not distinctive enough to be used directly by such methods. For this reason we define another Matching Game that ignores the information given by the descriptors and takes advantage of the rigidity constraint to be enforced in the surface registration problem. While this can sound counterintuitive, the main idea of this approach is to limit the use of the weak features to the selection of interest points and to use a more reliable global approach (that does not depend on descriptors) for the registration itself.

Given a set of model interest points  $M$  and a set of data interest points  $D$  we define the set of strategies for our Matching Game as all the possible correspondences between them:  $S = \{(a_1, a_2) | a_1 \in M \text{ and } a_2 \in D\}$ . Of course for practical reasons it is perfectly reasonable to limit the size of  $S$  by including only pairs that show similar descriptors.

Once  $S$  has been selected, our goal becomes to extract from it the largest subset that includes only correctly matched points: that is, strategies that associate a point in the model surface with the same point in the data surface. To enforce this we assign to each pair of strategies a payoff that is inversely proportional to a measure of violation of the rigidity constraint. This violation can be expressed in several ways, but since all the rigid transformations preserve Euclidean distances, we choose this property to express the coherence between strategies.

**Definition 1.** *Given a function  $\pi : S \times S \rightarrow \mathbb{R}^+$ , we call it a rigidity-enforcing payoff function if for any  $((a_1, a_2), (b_1, b_2))$  and  $((c_1, c_2), (d_1, d_2)) \in S \times S$  we have that  $\| |a_1 - b_1| - |a_2 - b_2| \| > \| |c_1 - d_1| - |c_2 - d_2| \|$  implies  $\pi((a_1, a_2), (b_1, b_2)) < \pi((c_1, c_2), (d_1, d_2))$ . In addition, if  $\pi((a_1, a_2), (b_1, b_2)) = \pi((b_1, b_2), (a_1, a_2))$ ,  $\pi$  is said to be symmetric.*

A rigidity-enforcing payoff function is a function that is monotonically decreasing with the absolute difference of the Euclidean distances between respectively the model and data points of the strategies compared. In other words, given two strategies, their payoff should be high if the distance between the model points is equal to the distance between the data points and it should decrease as the difference between such distances increases.

Further, if we want matching to be one-to-one, we must put an additional constraint on the payoffs, namely that mates sharing a point are incompatible.

**Definition 2.** *A rigidity-enforcing payoff function  $\pi$  is said to be one-to-one if  $a_1 = b_1$  or  $a_2 = b_2$  implies  $\pi((a_1, a_2), (b_1, b_2)) = 0$ .*

Given a set of strategies  $S$  and an enumeration  $O = \{1, \dots, |S|\}$  over it, a *mating game* is a non-cooperative game where the population is defined as a vector  $\mathbf{x} \in \Delta^{|S|}$  and the payoff matrix  $C = (c_{ij})$  is defined as  $c_{ij} = \pi(s_i, s_j)$ , where  $s_i, s_j \in S$  are enumerated by  $O$  and  $\pi$  is a symmetric one-to-one rigidity-enforcing payoff function. Intuitively,  $\mathbf{x}_i$  accounts for the percentage of the population that plays the  $i$ -th strategy. By using a symmetric one-to-one payoff function in a mating game we are guaranteed that ESS's will not include mates sharing either model or data nodes (see [12]). Moreover, a mating game exhibits some additional interesting properties.

**Theorem 1.** *Given a set of model points  $M$ , a set of data points  $D = TM$  that are exact rigid transformations of the points in  $M$ , and a set of strategies  $S \subseteq M \times D$  with  $(m, Tm) \in S$  for all  $m \in M$ , and a mating game over them with a payoff function  $\pi$ , the vector  $\hat{\mathbf{x}} \in \Delta^{|S|}$  defined as*

$$\hat{\mathbf{x}}_i = \begin{cases} 1/|M| & \text{if } s_i = (m, Tm) \text{ for some } m \in M; \\ 0 & \text{otherwise,} \end{cases}$$

*is an ESS and obtains the global maximum average payoff.*

This theorem states that when matching a surface with a rigidly transformed copy of itself the optimal solution (i.e., the population configuration that selects

all the strategies assigning each point to its copy) is the stable state of maximum payoff. Clearly, aligning a surface to an identical copy is not very useful in practical scenarios, where occlusion and measurement noise come into play. While the quality of the solution in presence of noise will be assessed experimentally, we can give some theoretical results regarding occlusions.

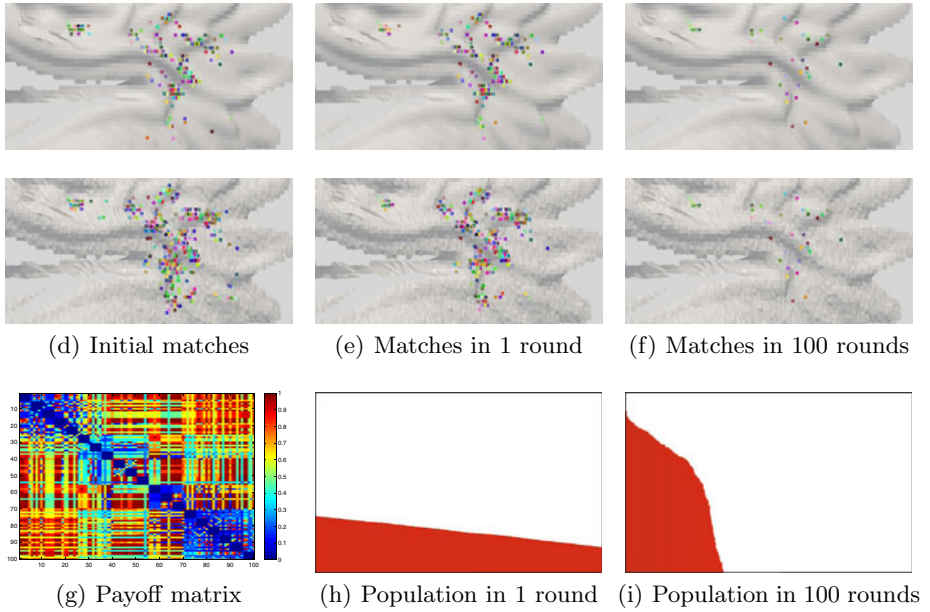
**Theorem 2.** *Let  $M$  be a set of points with  $M_a \subseteq M$  and  $D = TM_b$  a rigid transformation of  $M_b \subseteq M$  such that  $|M_a \cap M_b| \geq 3$ , and  $S \subseteq M_a \times D$  be a set of strategies over  $M_a$  and  $D$  with  $(m, Tm) \in S$  for all  $m \in M_a \cap M_b$ . Further, assume that the points that are not in the overlap, that is the points in  $E_a = M_a \setminus (M_a \cap M_b)$  and  $E_b = M_b \setminus (M_a \cap M_b)$ , are sufficiently far away such that for every  $s \in S, s = (m, Tm)$  with  $m \in M_a \cap M_b$  and every  $q \in S, q = (m_a, Tm_b)$  with  $m_a \in E_a$  and  $m_b \in E_b$ , we have  $\pi(q, s) < \frac{|M_a \cap M_b| - 1}{|M_a \cap M_b|}$ , then, the vector  $\hat{x} \in \Delta^{|S|}$  defined as*

$$\hat{x}_i = \begin{cases} 1/|M| & \text{if } s_i = (m, Tm) \text{ for some } m \in M_a \cap M_b; \\ 0 & \text{otherwise,} \end{cases}$$

is an ESS.

The result of theorem 2 is slightly weaker than theorem 1, as the face of the simplex corresponding to the “correct” overlap, while being an evolutionary stable state, is not guaranteed to obtain the overall highest average payoff. This is not a limitation of the framework as this weakening is actually due to the very nature of the alignment problem itself. The inability to guarantee the maximality of the average payoff is due to the fact that the original object ( $M$ ) could contain large areas outside the overlapping subset that are perfectly identical. Further, objects that are able to slide (for instance a plane or a sphere) could be allowed to move between different mixed strategies without penalty. These situations cannot be addressed by any algorithm without relying on supplementary information. However, in practice, they are quite unlikely extreme cases. In the experimental section we will show that our approach can effectively register a wide range of surface types.

In Fig. 3 we show a complete example of the evolutionary matching process. In order to make the example easy to understand we restricted our focus to a detail of a range scan of the Stanford “dragon”. In this example (and throughout all the experimental section)  $S$  is built by including all the strategy pairs composed by a feature point in the model and the 5 nearest feature points in the data in terms of Surface Hash (in this example we used an Integral Hash with 3 scales). In Fig. 3(g) we show, on a colored scale from 0 to 1, the payoff matrix of the rigid enforcing function used (which is discussed in the experimental section). Note that in the diagonal area of the matrix blocks of five strategies with reciprocal 0 payoff can be found: this is related to the way we built  $S$ . In fact we chose to include for each model point 5 candidates in the data and they are mutually non compatible as they share the same source point and we are looking for a one-to-one match. In the top and bottom half of Fig. 3(d) we can see respectively



**Fig. 3.** Example of a rigid enforcing payoff and of the evolution of the matching process

model and data feature points at the beginning of the matching process. After just one round of replicator dynamics we see that many outliers have been peeled off from the initial set  $S$ , but still some wrong matches are present. After 100 iterations only a few matches have been retained, but it is easy to see that they are extremely coherent. Finally, in Fig. 3(h) and Fig. 3(i) we show the (sorted) population histogram respectively after 1 and 100 iterations. The first histogram shows that all the strategies are still played by a sizeable amount of the population, while after 100 iterations most of the consensus is held by the few surviving matches.

## 4 Experimental Results

In this section we study the behavior of the proposed surface registration technique with respect to different Surface Hashes and scales. In addition we evaluate both the performance of the proposed feature descriptor with other matches and the quality of the alignment obtained by comparison with other pipelines. The rigidity-enforcing payoff function used throughout the experiments is defined as

$$\pi((a_1, b_1), (a_2, b_2)) = \frac{\min(|a_1 - a_2|, |b_1 - b_2|)}{\max(|a_1 - a_2|, |b_1 - b_2|)} \quad (2)$$

where  $a_1$ ,  $a_2$ ,  $b_1$  and  $b_2$  are respectively the two model (source) and data (destination) points in the compared mating strategies. The initial set of strategies

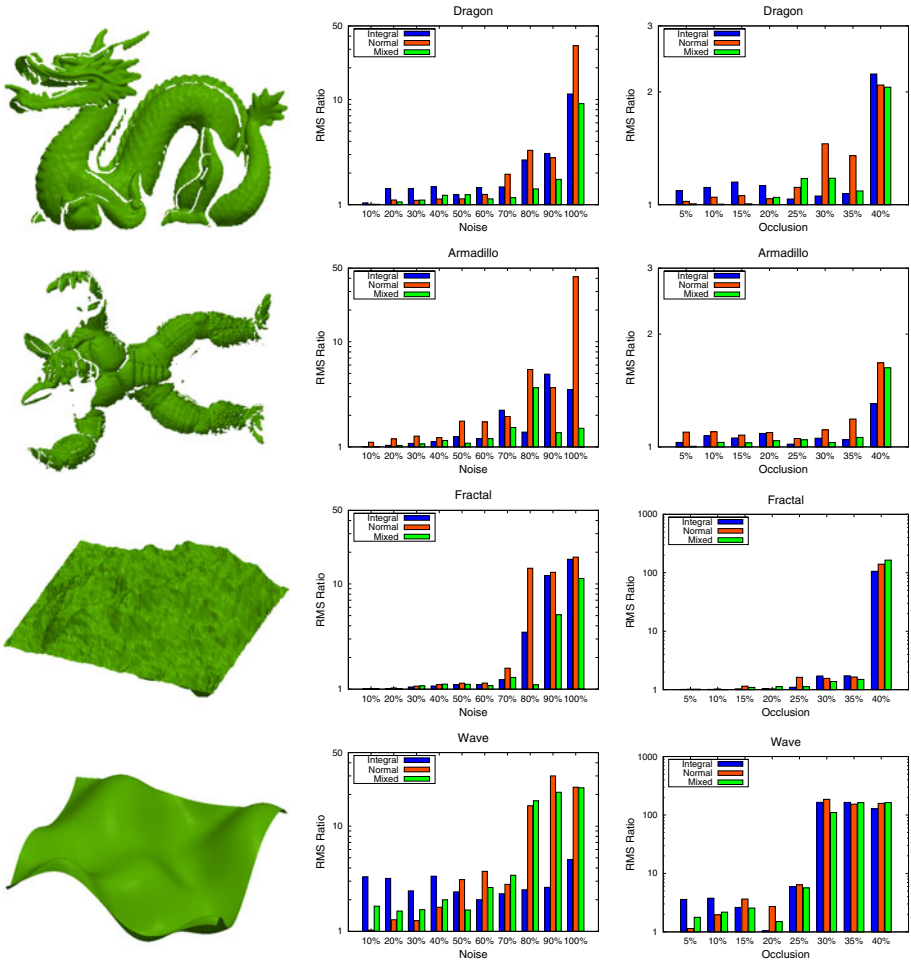
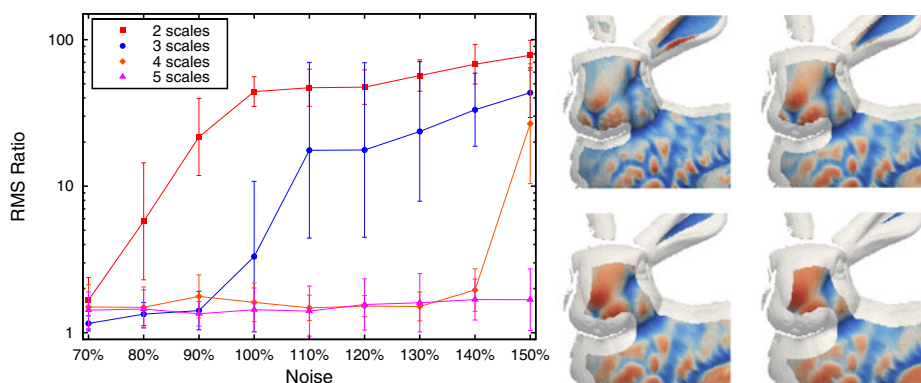


Fig. 4. Comparison of different descriptors using real and synthetic objects

$S$  was built by including all the pairs composed by a feature point in the model and the 5 feature points in the data with the nearest descriptor.

#### 4.1 Sensitivity to Noise, Occlusion, and Scale of the Descriptor

The performance of different descriptors was tested for various levels of noise and occlusion applied to two surfaces obtained from real range scans (“armadillo” and “dragon” from Stanford) and two synthetic surfaces designed to be challenging for coarse registration techniques (“fractal” and “wave”). The noise is a positional Gaussian perturbation on the point coordinates with its level ( $\sigma$ ) expressed in terms of the percentage of the average edge length, while the occlusion denotes the percentage of data and model surfaces removed. The RMS



**Fig. 5.** Effect of scale on the matching accuracy

Ratio in the charts is the ratio of the root mean square error (RMS) obtained after registration and the RMS of the ground truth alignment. The Normal and Integral Hashes were calculated over 3 levels of scale and the “Mixed” Hash is simply the juxtaposition of the previous two. In Fig. 4 we see that all the descriptors obtain good results with real ranges and the registration “breaks” only with very high levels of noise (on the same order of magnitude of the edge length). It is interesting to observe that the Mixed Hash always obtains the best performance, even with high level of noise: This higher robustness is probably due to the orthogonality between the Normal and Integral Hashes. The behavior with the “fractal” synthetic surface is quite similar, by contrast all the descriptors seem to perform less well with the “wave” surface. This is due to the lack of distinctive features on the model itself, which indeed represents a challenge for any feature based registration technique. The performance obtained with respect to occlusion is similar: all the descriptors achieve fairly good results and are resilient to high levels of occlusion (note that 40 percent occlusion is applied both to data and model). Overall the Mixed Hash appears to be consistently more robust. Since we found that the descriptors calculated over 3 levels of scale break at a certain level of noise, we were interested in evaluating if their performance can be improved by increasing their dimension. In Fig. 5 we present the results obtained with different levels of scale for the Mixed Hash. The graphs show the average over all the surfaces and the associated RMS. It is interesting to observe that by reducing the scale level the technique becomes less robust, whereas its performance increases dramatically when the number of scales increases. With a scale level of 5 our approach can deal even with surfaces subject to Gaussian positional noise of  $\sigma$  greater than the edge length. Unfortunately this enhanced reliability comes with a drawback: by using larger levels of scale the portion of boundary that cannot be characterized grows. In the right half of Fig. 5 the shrinking effect is shown for scale levels from 2 to 5.

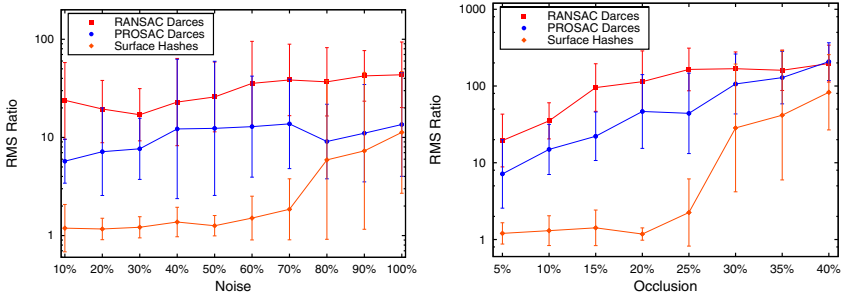


Fig. 6. Comparison of the performance obtained with different matchers

## 4.2 Comparisons with Other Matchers

Our goal in this set of experiments is to study if Surfaces Hashes can be used successfully with matchers alternative to the Matching Game described in Section 3.2. Specifically, we compared our full pipeline with standard DARCES [17] and with a DARCES variant that adopts PROSAC instead of plain RANSAC to take advantage of our descriptors as prior. To this end, we sorted the initial correspondence hypotheses by descriptor similarity and operated a PROSAC-like selection starting from an initial set of high-ranked matches and enlarging it progressively. In Fig. 6 we show the results of this test. As expected, RANSAC-based DARCES yields the worst results. Our PROSAC based variant obtains slightly better average registrations, but, the additional information provided by the descriptors is not distinctive enough to boost this technique to performance levels of the Matching Game that relies only on the global rigidity constraints.

## 4.3 System-Level Comparisons

Since our alignment approach does not need any initial estimate of the motion between surfaces, it can be classified as a coarse registration technique. For this reason we found appropriate to compare it with other widely used coarse registration methods. To this extent, we chose to use the Spin Images based approach proposed by Johnson [9] and the MeshDOG/MeshHOG combination suggested by Zaharescu [11]. The latter was selected because it adopts short descriptors very similar to the one proposed in this paper. In Fig. 7 we see that both techniques perform worse than the one based on Surface Hashes, even at low noise and occlusion levels. Surprisingly MeshDOG/MeshHOG obtains the worst results, probably because of the combination of a weak descriptor with a greedy matcher. Finally, we used the coarse registrations obtained with each approach to initialize a fine registration made with a best-of-breed ICP variant similar to the one proposed in [18]. Point selection is based on Normal Space Sampling [19], and point-surface normal shooting is adopted for finding correspondences, distant mates, candidates with back-facing normals, or matings established on the boundary of the mesh are rejected. In the leftmost plot of Fig. 8 we histogram the frequency of RMS ratio intervals obtained after the coarse registration. The histogram is based on bins of exponentially increasing

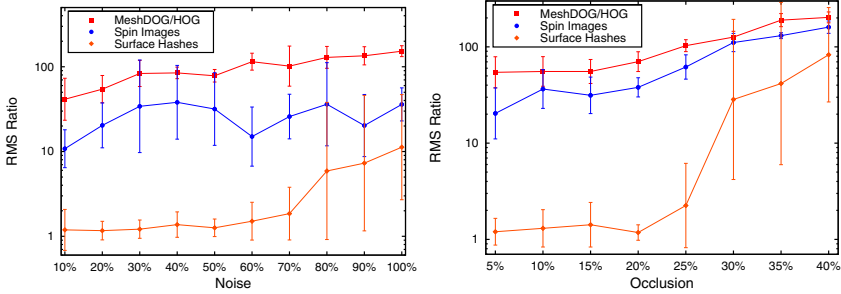


Fig. 7. Comparison of the performance obtained with different coarse techniques

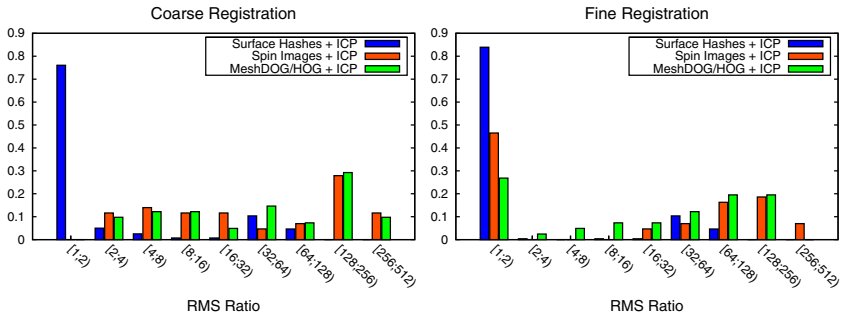


Fig. 8. Comparison of the performance between complete pipelines

size. In the rightmost chart the distribution change after a full round of ICP refinement can be seen. We can observe that while ICP is able to correct some wrong registrations with lower RMS Ratio, our approach still reaches the optimal alignment with a frequency that is almost double of the one obtained by the closest competitor. Regarding the computational complexity, it should be noted that the algorithm is quadratic in the number of strategies and thus the number of feature correspondences. Nevertheless, the initial interest points selection and the correspondences filtering by means of the descriptors, allow us to keep the computational time within a few seconds in all of our experiments.

## 5 Conclusions

In this paper we introduced a novel surface registration technique that uses very simple descriptors to create several weak correspondence hypotheses that are further optimized by a robust game-theoretic matcher. A theoretical result exposed the correspondence between optimal alignments and evolutionary equilibria, and the approach was validated on a wide range of experiments showing its greater robustness with respect to noise and occlusion in comparison with other well-known techniques.

## Acknowledgement

We acknowledge the financial support of the Future and Emerging Technology (FET) Programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open project SIMBAD grant no. 213250.



## References

1. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. Fourth Alvey Vision Conference, pp. 147–151 (1988)
2. Marr, D., Hildreth, E.: Theory of edge detection. Royal Soc. of London Proc. Series B 207, 187–217 (1980)
3. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing* 22, 761–767 (2004); *British Machine Vision Computing 2002* (2002)
4. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002, Part I. LNCS*, vol. 2350, pp. 128–142. Springer, Heidelberg (2002)
5. Lowe, D.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 20, 91–110 (2003)
6. Herbert Bay, T.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)
7. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 1615–1630 (2005)
8. Chua, C.S., Jarvis, R.: Point signatures: A new representation for 3d object recognition. *International Journal of Computer Vision* 25, 63–85 (1997)
9. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* 21, 433–449 (1999)
10. Pottmann, H., Wallner, J., Huang, Q.X., Yang, Y.L.: Integral invariants for robust geometry processing. *Comput. Aided Geom. Des.* 26, 37–60 (2009)
11. Zaharescu, A., Boyer, E., Varanasi, K., Horaud, R.P.: Surface feature detection and description with applications to mesh matching. In: *CVPR* (2009)
12. Albarelli, A., Rota Bulò, S., Torsello, A., Pelillo, M.: Matching as a non-cooperative Game. In: *ICCV. IEEE Computer Society, Los Alamitos* (2009)
13. Weibull, J.: *Evolutionary Game Theory*. MIT Press, Cambridge (1995)
14. Albarelli, A., Rodolà, E., Torsello, A.: A game-theoretic approach to fine surface registration without initial motion estimation. In: *CVPR* (2010)
15. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 381–395 (1981)
16. Chum, O., Matas, J.: Matching with prosac - progressive sample consensus. In: *CVPR*, Washington, DC, USA, pp. 220–226. IEEE Computer Society, Los Alamitos (2005)
17. Chen, C.S., Hung, Y.P., Cheng, J.B.: Ransac-based darces: A new approach to fast automatic registration of partially overlapping range images. *IEEE Trans. Pattern Anal. Mach. Intell.* 21, 1229–1234 (1999)
18. Turk, G., Levoy, M.: Zippered polygon meshes from range images. In: *SIGGRAPH 1994: Proc. of the 21st annual conference on Computer graphics and interactive techniques*, pp. 311–318. ACM, New York (1994)
19. Rusinkiewicz, S., Levoy, M.: Efficient variants of the icp algorithm. In: *Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, pp. 145–152 (2001)

# Accelerated Hypothesis Generation for Multi-structure Robust Fitting

Tat-Jun Chin, Jin Yu, and David Suter

School of Computer Science, The University of Adelaide, South Australia  
{tjchin, jin.yu, dsuter}@cs.adelaide.edu.au

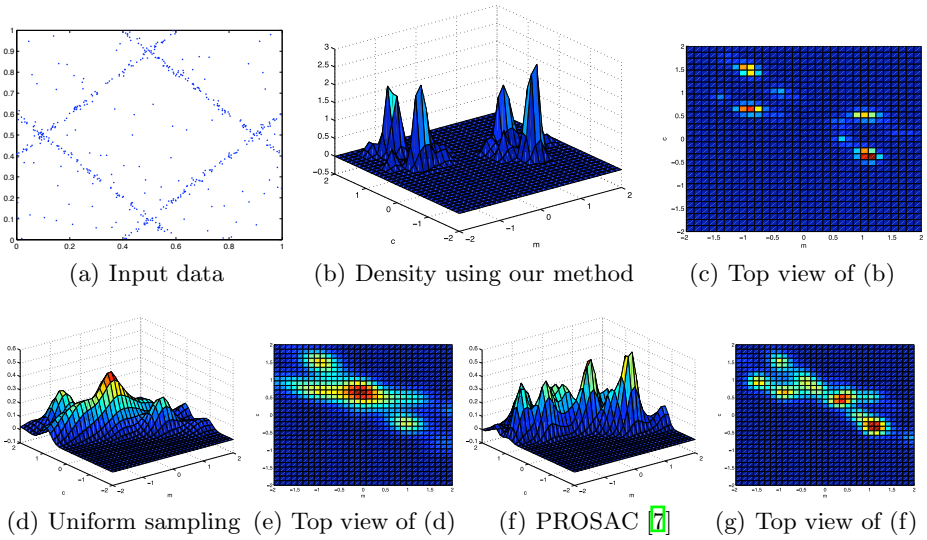
**Abstract.** Random hypothesis generation underpins many geometric model fitting techniques. Unfortunately it is also computationally expensive. We propose a fundamentally new approach to accelerate hypothesis sampling by guiding it with information derived from residual sorting. We show that residual sorting innately encodes the probability of two points to have arisen from the same model and is obtained without recourse to domain knowledge (*e.g.* keypoint matching scores) typically used in previous sampling enhancement methods. More crucially our approach is naturally capable of handling data with multiple model instances and excels in applications (*e.g.* multi-homography fitting) which easily frustrate other techniques. Experiments show that our method provides superior efficiency on various geometric model estimation tasks. Implementation of our algorithm is available on the authors’ homepage.

## 1 Introduction

Random hypothesis sampling is central to many state-of-the-art robust estimation techniques. The procedure is often embedded in the “hypothesise-and-verify” framework commonly found in methods such as Random Sample Consensus (RANSAC) [1] and Least Median Squares (LMedS) [2]. The goal of sampling is to generate many putative hypotheses of a given geometric model (*e.g.* fundamental matrix, homography) from randomly chosen minimal subsets of the input data. The hypotheses are then scored in the verification stage according to a robust criterion (*e.g.* number of inliers, median of squared residuals).

The underlying principle of random hypothesis sampling is to “hit” at least one all-inlier subset corresponding to a particular genuine instance of the geometric model. Unfortunately the total number of hypotheses required such that this happens with significant chance scales with the fraction of outlier contamination. For heavily contaminated data hypothesis generation easily becomes the computational bottleneck. Moreover in data with *multiple* instances of the geometric model (also called “structures” [3]) the inliers of one structure behave as *pseudo*-outliers to the other structures, thus further compounding the problem.

Due to the widespread usage of robust estimators in Computer Vision there have been many innovations [4–6] to speed-up random hypothesis generation. These methods aim to guide the sampling process such that the probability of hitting an all-inlier subset is improved. The trick is to endow each input datum



**Fig. 1.** Given the input data in (a) where there are 4 lines with 100 points per line and 100 gross outliers, we sample 100 line hypotheses using the proposed method, uniform sampling (à la the original RANSAC [1]) and PROSAC [7], yielding the parameter space density plotted respectively in (b), (d) and (f). Notice that the hypotheses of our method are concentrated on the correct models yielding 4 distinct peaks. Results from uniform sampling and PROSAC however contain many false peaks and irrelevant hypotheses. As Sec. 4 shows, in multi-structure data our method can successfully “hit” all true models using considerably less time than previous techniques.

a prior probability of being an inlier and to sample such that data that have high probabilities are more likely to be *simultaneously* selected. Such prior probabilities are often derived from domain-specific knowledge. For example Guided-MLESAC [6] and PROSAC [7] concentrate the sampling effort on correspondences with higher keypoint matching scores, the rationale being that inlier correspondences originate from confident keypoint matches (recall that in geometry estimation one correspondence consists of two matching points in different views). SCRAMSAC [9] further imposes a spatial consistency filter so that only correspondences which respect local geometry get sampled. Other works assume that inliers form dense clusters [5] or lie in meaningful image segments [8].

A crucial deficiency of previous methods lies in regarding the inlier probability of a datum to be *independent* of the other data. This is untrue when there are multiple structures. Given that an inlier of one structure is chosen, the probability that a second datum is an inlier (and thus should be chosen as well) depends on whether the second datum arose from the *same* structure. In other words, it is very possible that two correspondences with high keypoint matching scores are inliers from *different* valid structures. Methods that ignore this point are bound to wastefully generate many invalid *cross-structure* hypotheses. Our results on real and synthetic data (see Fig. 1) prove that this is indeed the case.

We also argue that the domain knowledge used in previous guided sampling techniques do not translate into convincing prior inlier probabilities. For example, inliers of a valid homography relation do not necessarily cluster together, while false or irrelevant correspondences can have high matching scores especially on scenes with repetitive textures. In the general case it is often questionable whether some usable and *reliable* domain knowledge is always available.

In this paper we propose a fundamentally novel technique to accelerate random hypothesis generation for robust model fitting. Our guided sampling scheme is driven only by residual sorting information and does not require domain- or application-specific knowledge. The scheme is encoded in a series of inlier probabilities which are updated on-the-fly. Most importantly our inlier probabilities are conditional on the selected data and thus encourages only inliers from the *same* structure to be simultaneously chosen for estimation. As our results demonstrate (see Sec. 4), our technique provides superior sampling efficiency especially on multi-structure data where other methods simply breakdown.

The rest of the paper is organised as follows: Sec. 1.1 surveys related work to put this paper in the right context. Sec. 2 describes the basic principles leading to our novel hypothesis generation scheme in Sec. 3. Sec. 4 outlines our experimental results and Sec. 5 draws conclusions.

## 1.1 Related Work

Many previous enhancements on the hypothesise-and-verify framework occur in the context of the RANSAC method. A recent survey [10] categorises them roughly into three groups. The first group of methods [4-9] aim to improve the random hypothesis sampling routine such that the chances of hitting an all-inlier sample is increased. In LO-RANSAC [4] an inner RANSAC loop is introduced into the main RANSAC body such that hypotheses may be generated from the set of inliers found so far, thus improving the consensus score more rapidly. Guided-MLESAC [6] and PROSAC [7] focus the sampling on more promising data based on keypoint matching scores, and this is extended to include spatial verification in SCRAMSAC [9]. In [5] sampling is concentrated on neighbouring correspondences, and in a similar spirit GroupSAC [8] focusses sampling on groups of data obtained using image segmentation. We emphasise that our work belongs to this category with the novelty of being domain-independent and optimised for accelerated hypothesis sampling in multi-structure data.

The second group of innovations [11-14] speed-up the hypothesis verification stage by minimising the time expended for evaluating unpromising hypotheses. The  $T_{d,d}$  test [11] evaluates a hypothesis on a small random subset of the input data. This may mistakenly reject good hypotheses thus a much larger number of samples are required. However the overall time can potentially be reduced since the verification now consumes less time. Bail-Out test [12] and WaldSAC [13, 14] respectively apply catch-and-release statistics and Wald's theory of sequential decision making to allow early termination of the verification of a bad hypothesis.

The third category [10, 15] considers RANSAC in a real-time setting. The goal is to find the best model from a fixed number of hypotheses afforded by

the allotted time interval. Given a set of hypotheses, Preemptive RANSAC [15] scores them in a breadth-first manner such that unpromising hypotheses can be quickly filtered out from the subsequent passes. ARRANSAC [10] performs a partially breadth-first verification such that the number of hypotheses may be modified according to the inlier ratio estimate while still bounding the runtime.

We are also aware of recent work [16, 17] that side-steps the hypothesise-and-verify framework and solves robust estimation directly as a global optimisation problem. While providing globally optimal solutions, these methods require considerably more time than RANSAC, especially for higher order geometric models. Our concern in this paper is to efficiently fit a geometric model onto noisy data with minimal loss to accuracy, and therefore our aims are different to [16, 17]. We also note that these methods [16, 17] currently cannot handle multi-structure data which make up a significant proportion of practical problems.

## 2 Inlier Probabilities from Sorting Information

We first describe how inlier probabilities can be derived from residual sorting information. Let  $\mathcal{X} := \{\mathbf{x}_i\}_{i=1}^N$  be a set of  $N$  input data. Under the hypothesise-and-verify framework, a series of tentative models (or hypotheses)  $\{\theta_1, \dots, \theta_M\}$  are generated from minimal subsets of the input data where  $M$  is the number of hypotheses generated. For each datum  $\mathbf{x}_i$  we compute its absolute residuals as measured to the  $M$  hypotheses to form the residual vector

$$\mathbf{r}^{(i)} := [ r_1^{(i)} \ r_2^{(i)} \ \dots \ r_M^{(i)} ]. \quad (1)$$

Note that the hypotheses do not lie in any particular order except the order in which they are generated. We then find the permutation

$$\mathbf{a}^{(i)} := [ a_1^{(i)} \ a_2^{(i)} \ \dots \ a_M^{(i)} ] \quad (2)$$

such that the elements in  $\mathbf{r}^{(i)}$  are sorted in non-descending order, *i.e.*,

$$p < q \implies r_{a_p^{(i)}}^{(i)} \leq r_{a_q^{(i)}}^{(i)}. \quad (3)$$

The sorting  $\mathbf{a}^{(i)}$  essentially ranks the  $M$  hypotheses according to the *preference* of  $\mathbf{x}_i$ ; the higher a hypothesis is ranked the more likely  $\mathbf{x}_i$  is an inlier to it.

Intuitively, two data  $\mathbf{x}_i$  and  $\mathbf{x}_j$  will share many common hypotheses at the top of their preference list  $\mathbf{a}^{(i)}$  and  $\mathbf{a}^{(j)}$  *if* they are inliers from the *same* structure. This is independent of whether  $\mathbf{x}_i$  and  $\mathbf{x}_j$  coexist in the same neighbourhood or whether they are correspondences with high keypoint matching scores.

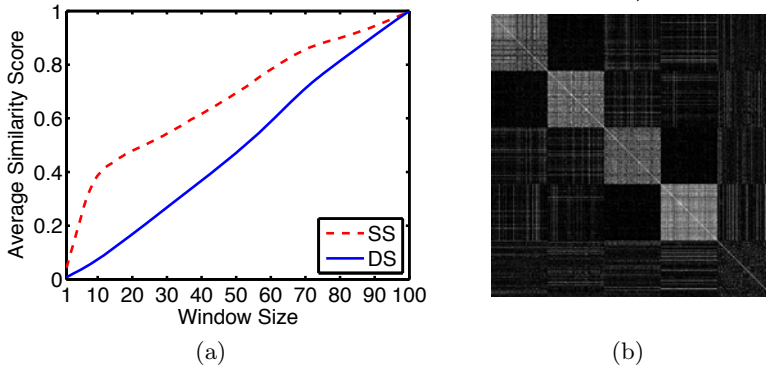
To illustrate this, first let  $\mathbf{a}_{1:h}^{(i)}$  be the vector with the first- $h$  elements of  $\mathbf{a}^{(i)}$ . We define the following function as the “intersection” between  $\mathbf{x}_i$  and  $\mathbf{x}_j$ :

$$f(\mathbf{x}_i, \mathbf{x}_j) := \frac{1}{h} \left| \mathbf{a}_{1:h}^{(i)} \cap \mathbf{a}_{1:h}^{(j)} \right|, \quad (4)$$

where  $|\mathbf{a}_{1:h}^{(i)} \cap \mathbf{a}_{1:h}^{(j)}|$  finds the number of identical elements shared by  $\mathbf{a}_{1:h}^{(i)}$  and  $\mathbf{a}_{1:h}^{(j)}$ . Window size  $h$  with  $1 \leq h \leq M$  specifies the number of leading hypotheses to take into account. Note that  $f(\mathbf{x}_i, \mathbf{x}_j)$  ranges between 0 and 1 and is symmetric with respect to its inputs. Also  $f(\mathbf{x}_i, \mathbf{x}_i) = 1$  for all  $i$ .

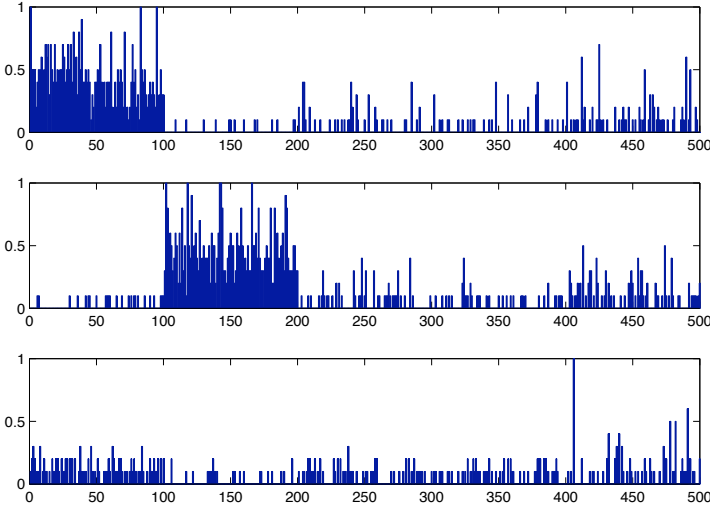
The window size  $h$  controls the discriminability of the intersection score given by (4). It is found empirically that across a wide range of  $h$  values this score is discriminative. For the data in Fig. 1(a) where  $M = 100$  we obtain the responses of  $f(\mathbf{x}_i, \mathbf{x}_j)$  while  $h$  is varied from 1,  $\dots$ ,  $M$ . Fig. 2(a) plots the mean of the responses which are separated according to whether the two input data are inliers from the same structure (denoted ‘‘SS’’) or otherwise (denoted ‘‘DS’’). The result clearly shows that inliers from the same structure have higher intersection values relative to other possible pairs of inputs. Based on the result in Fig. 2(a), we set  $h = \lceil 0.1 \times M \rceil$  by default for the intersection function unless mentioned otherwise.

We then obtain the  $N \times N$  matrix  $K$  where the element at the  $i$ -th row and  $j$ -th column is simply  $f(\mathbf{x}_i, \mathbf{x}_j)$ . Fig. 2(b) displays the matrix  $K$  by rearranging the points according to their structure membership, i.e.,  $\mathbf{x}_1$  to  $\mathbf{x}_{100}$  are inliers from structure 1,  $\mathbf{x}_{101}$  to  $\mathbf{x}_{200}$  are inliers from structure 2 and so on. The gross outliers are  $\mathbf{x}_{401}$  to  $\mathbf{x}_{500}$ . This makes visible a block diagonal pattern which confirms that strong mutual support occur among inliers of the same structure. We emphasise that such an arrangement is purely to aid in presentation and is unnecessary for  $f(\mathbf{x}_i, \mathbf{x}_j)$  or our subsequent steps to work.



**Fig. 2.** (a) Average intersection values for the data in Fig. 1(a) while  $h$  is varied. (b) Matrix  $K$  of size  $100 \times 100$  corresponding to  $h = 10$ .

We further analyse the results by plotting in Fig. 3 the values of selected rows of  $K$ . Unsurprisingly at a row corresponding to an inlier the significant values concentrate mostly on other inliers from the same structure, while for a gross outlier the values are generally low and appear to be randomly distributed. Therefore, given that a datum is selected, our idea is to use the intersection values of the datum as weights to sample a second datum. This yields inlier probabilities that encourages sampling within coherent structures. We emphasise that this phenomenon or idea is independent of the *type* of the geometric model.



**Fig. 3.** Values at rows of  $K$  corresponding to an inlier from structure 1 (Top), an inlier from structure 2 (Center) and a gross outlier (Bottom)

More generally the observations in Figs. 2 and 3 indicate considerable potential in residual sorting information. This concept has been pursued in constructing statistical learning-based model fitting [18] and robust clustering methods [19]. In the next section we illustrate how residual sorting can be exploited to drive a very efficient hypothesis generation scheme.

### 3 Guided Sampling for Multi-structure Data

We use the similarity function (4) to design a guided sampling scheme (Multi-GS, Algorithm 1) that is optimised for multi-structure data.

**The Weighting Function** Assume  $M$  model hypotheses have been generated so far and we wish to sample the next hypotheses (in a guided fashion). Let the model to be fitted be determined by a minimal subset  $\mathcal{S} := \{\mathbf{s}_k\}_{k=1}^p \subseteq \mathcal{X}$  of  $p$  data, where  $\mathbf{s}_k$  are indexed by the order in which they are sampled. The first datum  $\mathbf{s}_1$  is selected randomly. We then define a *basis weighting function*

$$w(\mathbf{x}_i, \mathbf{x}_j) := \begin{cases} f(\mathbf{x}_i, \mathbf{x}_j) & \text{if } \mathbf{x}_i \neq \mathbf{x}_j, \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where  $f$  is the intersection function (4). Given the first selection  $\mathbf{s}_1$ , the conditional probability  $P(\mathbf{x}_j|\mathbf{s}_1)$  of selecting  $\mathbf{x}_j$  as the second datum in the current minimal subset is then determined by the following monotonic relation:

$$w_1(\mathbf{x}_m) \geq w_1(\mathbf{x}_n) \implies P(\mathbf{x}_m|\mathbf{s}_1) \geq P(\mathbf{x}_n|\mathbf{s}_1), \tag{6}$$

where  $w_1(\cdot)$  is a *weighting function* conditioned on  $\mathbf{s}_1$

$$w_1(\cdot) := w(\cdot, \mathbf{s}_1). \quad (7)$$

The monotonic relation (6) says that data that are consistent with  $\mathbf{s}_1$  (according to the intersection score (4)) are more likely to be selected. Effectively  $w_1(\mathbf{x}_j)$  for  $j = 1, \dots, N$  is a set of *sampling weights* to choose the second datum.

The remaining members (i.e.,  $\mathbf{s}_3, \mathbf{s}_4, \dots, \mathbf{s}_p$ ) are also chosen *conditionally* on the data that are already drawn into the current minimal subset. Specifically the sampling weights for the  $(k + 1)$ -th member of the minimal subset is

$$w_k(\mathbf{x}_j) := \prod_{i=1}^k w(\mathbf{x}_j, \mathbf{s}_i). \quad (8)$$

This is the element-wise multiplication of the rows of matrix  $K$  (see Fig. 2(b)) corresponding to data that have already been selected  $\mathbf{s}_1, \dots, \mathbf{s}_k$ . The conditional probability  $P(\mathbf{x}_j | \mathbf{s}_1, \dots, \mathbf{s}_k)$  of selecting  $\mathbf{x}_j$  then follows from the rule

$$w_k(\mathbf{x}_m) \geq w_k(\mathbf{x}_n) \implies P(\mathbf{x}_m | \mathbf{s}_1, \dots, \mathbf{s}_k) \geq P(\mathbf{x}_n | \mathbf{s}_1, \dots, \mathbf{s}_k). \quad (9)$$

This continues until  $p$  data have been selected. The  $(M + 1)$ -th hypothesis is then estimated from the new minimal subset. Note that since (5) imposes  $w_k(\mathbf{s}_k) = 0$  a datum cannot be chosen more than once into the *same* minimal subset.

**Updating of Sampling Weights.** Theoretically the sampling weights (8) are updated as soon as a new hypothesis is produced since (5) uses all available

---

### Algorithm 1. Guided-Sampling for Multi-structure Robust Fitting (Multi-GS)

---

```

1: input input data  $\mathcal{X}$ , total number of hypotheses  $T$ , size of a minimal subset  $p > 0$ 
   and block size  $b > 0$ .
2: output a set  $\Theta$  of  $T$  model hypotheses.
3: for  $t := 1, 2, \dots, T$  do
4:   if  $t \leq b$  then
5:     randomly sample  $p$  data and store as  $\mathcal{S}$ 
6:   else
7:     select a datum  $\mathbf{s}_1$  from  $\mathcal{X}$  and initialise  $\mathcal{S} := \{\mathbf{s}_1\}$ 
8:     for  $k := 1, 2, \dots, (p - 1)$  do
9:       sample  $\mathbf{s}_{k+1}$  from  $\mathcal{X}$  by following the rule (9)
10:       $\mathcal{S} := \mathcal{S} \cup \{\mathbf{s}_{k+1}\}$ 
11:     end for
12:   end if
13:    $\Theta := \Theta \cup \{\text{Hypothesis instantiated from } \mathcal{S}\}$ 
14:   if  $t > b$  and  $\text{mod}(t, b) == 0$  then
15:     update the permutation  $\mathbf{a}^{(i)}$  (2) for all data
16:   end if
17: end for
18: return  $\Theta$ 

```

---



hypotheses. From a computational standpoint this is inefficient because a single new hypothesis does not add much information about inlier probabilities. Our proposed algorithm thus updates the weighting function only after a block (of size  $b$ ) of new hypotheses are generated; see Step 14–16 in Algorithm 1.

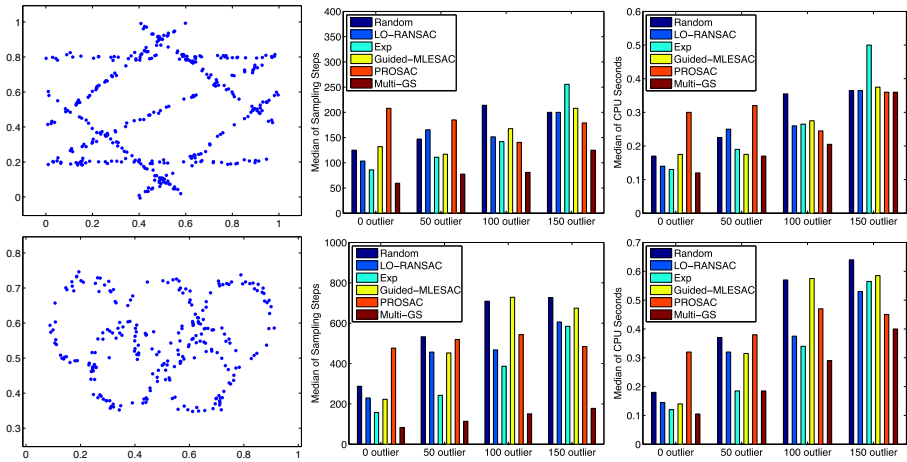
The weighting function is practically updated by modifying the permutations  $\mathbf{a}^{(i)}$  to account for the new hypotheses. We propose an efficient strategy to perform this step. Firstly, assume that we have the absolute residuals  $\{\mathbf{r}^{(i)}\}_{i=1}^N$  for the  $M$  hypotheses sampled so far. Each of these is sorted increasingly to obtain the permutation vectors  $\{\mathbf{a}^{(i)}\}_{i=1}^N$  of which only the top- $h$  elements  $\{\mathbf{a}_{1:h}^{(i)}\}_{i=1}^N$  partake in the computation of the sampling weights. The key to efficient updating is to fully retain  $\{\mathbf{r}^{(i)}\}_{i=1}^N$  and  $\{\mathbf{a}^{(i)}\}_{i=1}^N$ . After  $b$  new hypotheses become available their absolute residuals to the dataset are computed and inserted using *binary search* into the sorted  $\{\mathbf{r}^{(i)}\}_{i=1}^N$ . The new leading hypotheses  $\{\mathbf{a}_{1:h}^{(i)}\}_{i=1}^N$  are then extracted from the updated sorting. A binary search insertion into a vector of length  $M$  scales as  $O(\log M)$  and we have  $b$  of these per datum. Therefore the total cost of updating  $\mathbf{a}^{(i)}$  for  $N$  data is  $O(Nb \log M)$ .

On the surface it seems that the somewhat higher computational cost constitutes a weakness. However our algorithm conducts a more informed sampling given a unit of time in comparison to other techniques. The result is that we require less total CPU time to hit at least one all-inlier subset of *all* valid structures in the data; this is validated by our experiments in Sec. 4. In contrast the other methods are much slower because they unproductively generate many invalid cross-structure hypotheses. In single structure data the proposed algorithm performs comparably to other guided sampling techniques.

## 4 Experiments

We evaluated the performance of the proposed method (Multi-GS, Algorithm 1) on both synthetic and real image datasets. We compared against other state-of-the-art sampling enhancement schemes: LO-RANSAC [4], proximity sampling [3, 20] (denoted “Exp”), Guided-MLESAC [6], and PROSAC [7]. Uniform random sampling as in the original RANSAC [1] (denoted “Random”) is used as the baseline. We implemented all algorithms in MATLAB. All experiments were run on a Linux machine with 2.67GHz Intel quad core processors and 4 GB of RAM.

In all experiments the inlier threshold required by LO-RANSAC was set to the average residual of inliers as measured to their corresponding structures;  $T_N$  in PROSAC was set to  $5 \times 10^4$ . The scale parameter of Exp ( $\sigma^2$  as in Equation 1 in [20]) was set to twice the squared average nearest neighbour distance. We implemented Guided-MLESAC such that data points with higher quality scores (*e.g.* keypoint matching scores) are given higher probabilities to be drawn. For our method we consistently fixed the block size  $b = 10$  and the window size  $h = \lceil 0.1 \times t \rceil$ ,  $t$  being the number hypotheses generated so far.



**Fig. 4.** The performance of various sampling methods on 2D geometric data. The left column shows the generated inliers. The centre and right columns respectively show the median number of sampling steps and total CPU seconds needed to hit at least one all-inlier subset in each structure as the number of gross outliers is varied.

#### 4.1 Multiple Line and Circle Fitting

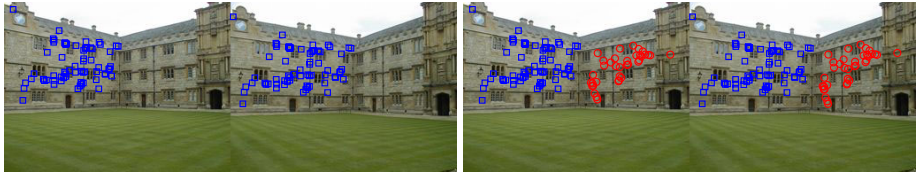
We first compare our algorithm against previous methods on multiple line and circle fitting in 2D. Fig. 4 (left column) depicts the synthetically generated inliers (respectively 7 lines and 5 circles). We also add random gross outliers to increase the difficulty of the problem. The inlier noise scale and the number of inliers per structure were fixed to  $10^{-2}$  and 50, respectively. PROSAC and Guided-MLESAC require each datum to be associated with a quality score; we simulate this by probabilistically assigning inliers with higher scores than gross outliers.

Each method is given 20 random runs. The centre and right columns in Fig. 4 respectively show the median of sampling steps (i.e., number of hypotheses) and total CPU seconds required to recover *at least* one all-inlier minimal subset for each structure *vs* the number of gross outliers in the data. It can be seen that our method is the most efficient in terms of the required sampling steps for both lines and circles. For instance, in the case of circle fitting, Multi-GS typically takes no more than half of the sampling steps needed by the other methods. In terms of total time expended Multi-GS still require less CPU seconds than the others, especially so for circle fitting. This suggests that the performance gap between our method and previous approaches would widen for higher order geometric models. Indeed, we demonstrate this in the next two experiments.

#### 4.2 Homography Estimation

Our second set of experiments involve estimating planar homographies on real image data.<sup>1</sup> Putative keypoint correspondences and their corresponding scores

<sup>1</sup> <http://www.robots.ox.ac.uk/~vgg/data>



(a) Left: College-1. Right: College-2.



(b) Left: College-3. Right: College-4.

**Fig. 5.** Images pairs used in the experiments of Sec. 4.2 with marked *inlier* keypoints

were obtained by SIFT matching [21, 2] Fig. 5 shows the image pairs used in our experiments with marked keypoints. Note that for clarity we show only the *true inliers* in Fig. 5; there actually exist a large number of false correspondences (ranging between 20 to 100 depending on the images) which represent gross outliers due to incorrect SIFT matches. We use 4 correspondences to estimate a homography via Direct Linear Transformation (DLT, [22]). For each method, 50 random runs were performed, each for 60 CPU seconds.

Our experiments start with a relatively easy task which involves estimating a single homography for the planar structure marked in the College-1 image pair (Fig. 5, top left). The data contain 70 inlier correspondences with an approximately 41% inlier ratio. As can be seen in Table 1, by leveraging the SIFT matching scores, PROSAC hits an all-inlier sample at the very first iteration, costing nearly zero CPU time, whereas Exp and Guided-MLESAC perform better than the others in terms of the total number of all-inlier samples found within the given time budget. Overall, the performance of all methods are comparable on this simple *single structure* recovery task.

We now move to the setting of multi-structure fitting. The performance of various sampling methods was evaluated on the image pairs that contain 2–4 planar structures (Fig. 5, top right and bottom rows). Table 1 shows that the proposed Multi-GS is superior in terms of CPU seconds required to hit at least one all-inlier subset from all structures. For instance, on College-3 and College-4, Multi-GS requires around 80% *less* time than the best performing competing method. Moreover, within the given CPU time limit the total number of all-inlier subsets found by our method is typically much more than that of other methods.

<sup>2</sup> We used the code given on <http://www.vlfeat.org/~vedaldi/code/sift.html>

**Table 1.** Performance of various sampling methods over 50 random runs, each for 60 CPU seconds. We report the median of CPU time (CPU) (*resp.* sampling steps (Iter)) that is required to find at least one all-inlier minimal subset for each structure present in the data in Fig. 5. The average number of all-inlier samples found within 60 CPU seconds is listed separately for each structure I- $i$ ,  $i = 1, 2, \dots$ . The number of inliers and the inlier ratio for each I- $i$  is given in the parenthesis. The top result with respect to each performance measure are boldfaced.

Data		Random	LO-RAN SAC	Exp	Guided- MLESAC	PROSAC	Multi- GS
College-1	CPU	0.06	0.02	0.02	0.01	$< 10^{-3}$	0.02
	Iter	33	13	7	7	<b>1</b>	13
	I-1 (70, 41%)	978	1012	2782	<b>3591</b>	1380	1119
College-2	CPU	1.39	0.47	0.65	0.39	0.14	<b>0.11</b>
	Iter	836	261	354	229	77	<b>47</b>
	I-1 (70, 34%)	450	438	<b>1183</b>	1164	576	867
	I-2 (36, 17%)	30	29	63	96	43	<b>350</b>
College-3	CPU	1.03	0.75	0.62	0.67	0.69	<b>0.14</b>
	Iter	592	418	336	374	374	<b>54</b>
	I-1 (71, 22%)	72	71	140	134	96	<b>292</b>
	I-2 (80, 24%)	116	115	266	401	166	<b>494</b>
	I-3 (78, 24%)	101	100	199	82	88	<b>286</b>
College-4	CPU	5.23	5.34	2.42	3.49	1.62	<b>0.31</b>
	Iter	3060	3105	1309	2029	897	<b>113</b>
	I-1 (42, 15%)	18	17	47	29	24	<b>160</b>
	I-2 (42, 15%)	17	17	37	49	25	<b>171</b>
	I-3 (47, 17%)	28	27	57	60	39	<b>237</b>
	I-4 (42, 15%)	15	17	42	26	19	<b>91</b>

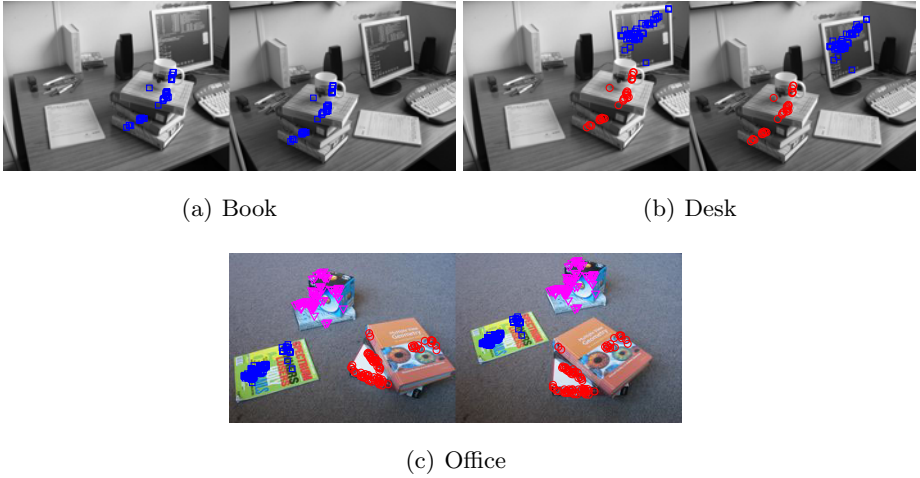
### 4.3 Fundamental Matrix Estimation

We also applied our sampling method to accelerate the estimation of fundamental matrices. Images of multiple moving objects were obtained from the web.<sup>3</sup> The keypoint correspondences and matching scores were obtained by SIFT matching. Hypotheses were generated from 7 keypoint correspondences via the standard 7-point estimation method [23].<sup>4</sup> For each method, 50 random runs were performed, each for 60 CPU seconds. Table 2 summarizes the performance of all methods on the three image pairs in Fig. 6. Again note that for clarity we only show the true inliers in Fig. 6. There exist from incorrect SIFT matchings many false correspondences which constitute gross outliers in the data.

Similar to the previous set of experiments, existing sampling methods are effective in sampling from single-structure data (*cf.* results on the Book data in Table 2), while they fail disastrously when more than one structure is present. Along with their inability to distinguish keypoint correspondences from different

<sup>3</sup> <http://www.iu.tu-darmstadt.de/datasets>

<sup>4</sup> <http://www.robots.ox.ac.uk/~vgg/hzbook/code/>



**Fig. 6.** Image pairs used in the experiments of Sec. 4.3 with marked *inlier* keypoints

**Table 2.** Performance of various sampling methods on image pairs in Fig. 6 (50 random runs with 60 CPU seconds per run). The same notations as used in Table 1 are used. In addition, we record the number of times a method fails to find at least one all-inlier sample for each structure within the given 60 CPU seconds (Fail). The reported median of CPU time (*resp.* sampling steps) is taken over successful runs only.

Data		Random	LO-RAN SAC	Exp	Guided- MLESAC	PROSAC	Multi- GS
Book	CPU	0.03	0.01	0.01	$< 10^{-3}$	$< 10^{-3}$	0.02
	Iter	11	4	4	3	<b>1</b>	8
	I-1 (28, 58%)	1426	1429	5879	<b>9152</b>	1778	184
	Fail	0	0	0	0	0	0
Desk	CPU	19.13	18.44	24.61	7.45	17.41	<b>0.18</b>
	Iter	5716	5572	7604	2983	5458	<b>41</b>
	I-1 (48, 27%)	7	7	79	40	8	<b>355</b>
	I-2 (28, 16%)	1	1	2	5	1	<b>175</b>
Fail	47	47	15	2	44	<b>0</b>	
Office	CPU	40.48	44.56	9.9	16.23	46.83	<b>0.17</b>
	Iter	13883	15442	3456	5719	15861	<b>38</b>
	I-1 (81, 24%)	2	2	10	6	2	<b>167</b>
	I-2 (78, 23%)	2	2	9	3	2	<b>234</b>
	I-3 (84, 24%)	2	3	10	6	3	<b>193</b>
Fail	35	22	0	7	23	<b>0</b>	

structures, the increase in the size of the minimal subset (from 4 in homography estimation to the current 7) makes the sampling from multi-structure data an extremely challenging task for previous methods. For instance, random sampling, LO-RANSAC, and PROSAC fail to find an all-inlier subset for each structure

in 44%-94% of the 50 random runs. As can be seen in Table 2, Multi-GS dramatically outperforms other methods in terms of all performance measures on the two multi-structure data: Desk and Office. It hits at least an all-inlier subset in each structure over an order of magnitude faster in terms of both CPU time and sampling steps. Moreover, within the given time limit, the overall all-inlier subsets found by our methods are up to over two orders of magnitude more than that obtained by other methods.

## 5 Conclusions

We propose a fundamentally new approach to accelerate hypothesis generation by guiding information derived from residual sorting. In contrast to existing sampling techniques, our approach is naturally capable of handling data with multiple structures. We also do not require potentially confusing domain knowledge needed by other techniques. We demonstrated, and compared our method on various multi-structure geometric modelling tasks. Our results show that the proposed method significantly outperforms previous techniques in terms of the total CPU time required to recover all valid structures in multi-structure data.

## Acknowledgements

This work is supported by the Australian Research Council grant DP0878801.

## References

1. Fischler, M.A., Bolles, R.C.: RANSAC: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM* 24, 381–395 (1981)
2. Rousseeuw, P.J., Leroy, A.M.: *Robust regression and outlier detection*. Wiley, Chichester (1987)
3. Stewart, C.V.: Robust parameter estimation in Computer Vision. *SIAM Review* 41, 513–537 (1999)
4. Chum, O., Matas, J., Kittler, J.: Locally optimized RANSAC. In: Michaelis, B., Krell, G. (eds.) *DAGM 2003*. LNCS, vol. 2781, pp. 236–243. Springer, Heidelberg (2003)
5. Kanazawa, Y., Kawakami, H.: Detection of planar regions with uncalibrated stereo using distributions of feature points. In: *BMVC* (2004)
6. Tordoff, B.J., Murray, D.W.: Guided-MLESAC: Faster image transform estimation by using matching priors. *TPAMI* 27, 1523–1535 (2005)
7. Chum, O., Matas, J.: Matching with PROSAC- progressive sample consensus. In: *CVPR* (2005)
8. Ni, K., Jin, H., Dellaert, F.: GroupSAC: Efficient consensus in the presence of groupings. In: *ICCV* (2009)
9. Sattler, T., Leibe, B., Kobbelt, L.: SCRAMSAC: Improving RANSAC's efficiency with a spatial consistency filter. In: *ICCV* (2009)

10. Raguram, R., Frahm, J.M., Pollefeys, M.: A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 500–513. Springer, Heidelberg (2008)
11. Matas, J., Chum, O.: Randomized RANSAC with  $t_{d,d}$  test. In: Image and Vision Computing (2004)
12. Capel, D.: An effective bail-out test for RANSAC consensus scoring. In: BMVC (2005)
13. Matas, J., Chum, O.: Randomized RANSAC with sequential probability ratio test. In: ICCV (2005)
14. Chum, O., Matas, J.: Optimal randomized RANSAC. TPAMI 30, 1472–1482 (2008)
15. Nister, D.: Preemptive RANSAC for live structure and motion estimation. In: ICCV (2003)
16. Enqvist, O., Kahl, F.: Two view geometry estimation with outliers. In: BMVC (2009)
17. Li, H.: Consensus set maximization with guaranteed global optimality for robust geometry estimation. In: ICCV (2009)
18. Chin, T.J., Wang, H., Suter, D.: Robust fitting of multiple structures: The statistical learning approach. In: ICCV (2009)
19. Chin, T.J., Wang, H., Suter, D.: The ordered residual kernel for robust motion subspace clustering. In: NIPS (2009)
20. Toldo, R., Fusiello, A.: Robust multiple structures estimation with j-linkage. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 537–547. Springer, Heidelberg (2008)
21. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
22. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2003)
23. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004) ISBN: 0521540518

# Aligning Spatio-Temporal Signals on a Special Manifold

Ruonan Li and Rama Chellappa

Center for Automation Research, University of Maryland  
College Park, MD, 20742, USA  
{liruonan, rama}@umiacs.umd.edu

**Abstract.** We investigate the spatio-temporal alignment of videos or features/signals extracted from them. Specifically, we formally define an *alignment manifold* and formulate the alignment problem as an optimization procedure on this non-linear space by exploiting its intrinsic geometry. We focus our attention on semantically meaningful videos or signals, *e.g.*, those describing or capturing human motion or activities, and propose a new formalism for temporal alignment accounting for executing rate variations among realizations of the same video event. By construction, we address this static and deterministic alignment task in a dynamic and stochastic manner: we regard the search for optimal alignment parameters as a recursive state estimation problem for a particular dynamic system evolving on the alignment manifold. Consequently, a Sequential Importance Sampling iteration on the alignment manifold is designed for effective and efficient alignment. We demonstrate the performance on several types of input data that arise in vision problems.

## 1 Introduction

In this paper, we consider the problem of aligning two spatio-temporal signals (*i.e.*, videos, their filtered versions, or spatio-temporal features extracted from them.) which come from the same dynamic scene or the same category of dynamics. The misalignment between the two signals, captured by distinct cameras at the same time or by the same camera at different times, may result from the differences in view points, view angles, internal calibration parameters, as well as temporal shifts and scaling. Previous work on video sequence alignment mostly used feature-based approaches [1-7] or direct approaches [8-10]. In the former class, features like two-frame correspondences of interest points or trajectories of tracked objects were used as inputs to the alignment algorithm, while in the latter, intensity, color, or other pixel/patch level appearance attributes were used. The spatial aspect of the misalignment was mostly modeled as one of the transforms including affine, homography, and perspective ones between the image plane coordinates of the two signals, based on different assumptions made regarding imaging conditions. The temporal misalignment, on the other



hand, mainly took frame rate and shift synchronization into account, modeled as a 1-D affine transform along the time axis. The algorithms were designed for parametric representations of the particular transforms to achieve optimal alignments. The warping parameters were then obtained using a numerical optimization method which is typically an exhaustive search or a greedy method such as gradient descent.

The first step taken in this work is to revisit the issue of temporal misalignment, which comes not only from the camera aspect (frame-rate and temporal shift), but also from the observed dynamics. We look into semantically meaningful visual dynamics beyond plain spatio-temporal volumes: one of the examples of semantically meaningful visual signals is videos recording human actions/activities. The same class of activities (*e.g.*, walking) may contain realizations executed at varying rates, though the essential characterization for that activity category is rate independent. This rate change is in fact a temporal misalignments among realizations (signals) and is described by a non-affine time warping [11, 12]. Therefore, a complete description of the temporal misalignment regarding these signals should include time warping as well. A second concern is about the spatial aspect of the alignment algorithm, which usually pertains particularly to either feature-based methods or direct methods and sticks to the parametric spatial transform assumed. Existing algorithms are far from being scalable and flexible to easily adapt to different parametric model and different inputs. Moreover, it is always crucial to strike a balance between computational complexity and convergence towards global optimum.

Taking all these factors into account, we reformulate the spatio-temporal alignment problem and provide a general framework and associated computational algorithms. Specifically, we propose the concept of the *alignment manifold*, which is the nonlinear space of all possible spatio-temporal transformations with an intrinsic geometric characterization. We detail the construction of the alignment manifold and discuss basic manipulations of the elements on it. The spatio-temporal signal alignment, consequently, becomes an optimization procedure on the manifold, regardless of whether the inputs are features or appearances, provided that an objective function is properly defined to measure the misalignment of the two signal under a spatio-temporal transformation model. In particular, we present a Bayesian optimization algorithm on the manifold based on Sequential Importance Sampling (SIS) [13], to achieve both efficiency and better convergence to the global optimum. The key idea is to regard the optimal alignment as a static state to be recursively estimated from the observed misalignment such that the posterior probability density of the estimated state reaches maximum at the true optimal alignment.

In short, the contributions of this paper are (1) we present a general framework for spatio-temporal alignment, incorporating temporal warping and various parametric spatial transforms as well as inputs; (2) we introduce the *alignment manifold*, a manifold tuned to the alignment task; and (3) a SIS algorithm is specifically designed for the alignment manifold to generate the numerical solution to the alignment problem.

## 2 The Framework of Alignment Problem

Given 3-dimensional spatio-temporal signals  $S^1$  and  $S^2$ , whose elements are denoted as  $S^1(x, y, t)$  and  $S^2(x, y, t)$  respectively, the spatio-temporal alignment problem aims to solve the following optimization problem

$$\min_{\mathbf{p} \in \mathfrak{M}} J(S^1, S^2, \mathbf{p}) \tag{1}$$

where  $\mathbf{p}$  is the parameter vector specifying the alignment transform,  $\mathfrak{M}$  is the alignment manifold, *i.e.*, the space of all feasible  $\mathbf{p}$ 's, and  $J$  is a measure of misalignment to be minimized by an optimal  $\mathbf{p}$ . As in previous efforts, we assume the relative internal and external parameters of the two cameras to be fixed but unknown, *i.e.*, both stationary or jointly moving. As a result, the spatial misalignment and temporal misalignment become decoupled. In other words, we may split  $\mathbf{p}$  into two components as  $\mathbf{p} = [\mathbf{p}_S^T, \mathbf{p}_T^T]^T$ , so that the spatial and temporal misalignment can be independently handled. (Cameras with relative motion and coupled spatio-temporal misalignment are important situations though beyond the scope of this work.) The alignment manifold  $\mathfrak{M}$  is accordingly decomposed into the Cartesian product of two submanifolds as  $\mathfrak{M} = \mathfrak{M}_S \times \mathfrak{M}_T$ , where  $\mathbf{p}_S \in \mathfrak{M}_S$  and  $\mathbf{p}_T \in \mathfrak{M}_T$ . The explicit analytical form of  $J$  depends on the specific spatial and temporal transform involved, as well as the measure of misalignment. We give three examples for illustrative purposes.

**Example 1.**  $S^1$  and  $S^2$  are grey-level videos, the spatial displacement is 2-D affine, and temporal transform is 1-D affine. The misalignment is measured as the pixel-wise mean square error. In this case,  $J(S^1, S^2, \mathbf{p}) = \sum_{x,y,t} (S^1(x, y, t) - S^2(x + u, y + v, t + w))^2$ , and

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 & b_1 \\ a_{21} & a_{22} & 0 & b_2 \\ 0 & 0 & a & b \end{bmatrix} \begin{bmatrix} x \\ y \\ t \\ 1 \end{bmatrix}. \tag{2}$$

The corresponding alignment parameter vectors are  $\mathbf{p}_S = (a_{11}, a_{12}, a_{21}, a_{22}, b_1, b_2)^T$  and  $\mathbf{p}_T = (a, b)^T$  with  $\mathfrak{M}_S$  to be the 2-D affine group  $\mathbb{A}(2)$  and  $\mathfrak{M}_T$  to be  $\mathbb{R}^+ \times \mathbb{R}$ .

**Example 2.**  $S^1$  and  $S^2$  are color videos, *i.e.*,  $S^i$  contain three channels  $S_j^i, j = 1, 2, 3$ , spatial transform is 2-D homography, and the temporal transform is a non-linear time warping. The misalignment is measured as the pixel-wise mean square error of the intensity. In this case,  $J(S^1, S^2, \mathbf{p}) = \sum_j \alpha_j \sum_{x,y,t} (S_j^1(x, y, t) - S_j^2(x', y', t'))^2$ ,  $x' = \frac{h_{11}x+h_{12}y+h_{13}}{h_{31}x+h_{32}y+h_{33}}$ ,  $y' = \frac{h_{21}x+h_{22}y+h_{23}}{h_{31}x+h_{32}y+h_{33}}$ , and  $t' = W(t)$ , where  $\alpha_j$ 's are the weights for the channels and  $W(t)$  is the time warping function. If we denote  $H = [h_{i,j}]_{3 \times 3}$  to be the homography matrix with the constraint of unit determinant (*i.e.*  $\det H = 1$ , without loss of generality), then we have  $\mathbf{p}_S = H$ ,  $\mathbf{p}_T = W$ ,  $\mathfrak{M}_S$  is the  $3 \times 3$  special linear group  $\mathbb{SL}(3)$ , and  $\mathfrak{M}_T$  is the set of all possible time warpings.

**Example 3.**  $S^1$  and  $S^2$  contain  $N$  spaces-time point trajectories respectively, *i.e.*,  $S^i = \{T_j^i\}_{j=1,2,\dots,N}$  and  $T_j^i = \{(x_j^i(t), y_j^i(t))\}_t$ , where  $(x_j^1(t), y_j^1(t))$  ( $x_j^2(t'), y_j^2(t')$ ) are assumed to come from the  $j$ th tracked interest point corresponding to the same 3-D point, captured by two pinhole cameras. Then considering perspective misalignment of the trajectories we have  $J(S^1, S^2, \mathbf{p}) = \sum_j \sum_t \|[x_j^1(t), y_j^1(t), 1]\mathbf{F}[x_j^2(W(t)), y_j^2(W(t)), 1]^T\|^2$ . Here  $\mathbf{F}$  is the  $3 \times 3$  fundamental matrix, and we may regard  $\mathbf{p}_S = \mathbf{F}$  and  $\mathfrak{M}_S$  to be the set of all possible fundamental matrices.

### 3 The Alignment Manifold

In this section we look into the alignment manifold  $\mathfrak{M} = \mathfrak{M}_S \times \mathfrak{M}_T$ , whose elements characterize the alignment transforms under consideration. As the spatial and temporal factors are considered independently in this work, we are in a position to discuss them separately.

#### 3.1 The Spatial Alignment Submanifold

The previous examples imply that the spatial alignment manifold  $\mathfrak{M}_S$  is usually identical to a Riemannian manifold of the transformation/constraint matrices. Affine group  $\mathbb{A}(2)$  and special linear group  $\mathbb{SL}(3)$  both belong to the *matrix Lie group*, which possesses several intrinsic geometric properties. We list a few used in this work: the geodesic (intrinsic) distance between two elements  $\mathbf{V}_1, \mathbf{V}_2$  on the matrix Lie group is  $d(\mathbf{V}_1, \mathbf{V}_2) = \|\log(\mathbf{V}_1^{-1}\mathbf{V}_2)\|$ . The exponential map  $\mathcal{E}_{\mathbf{V}_m} : \mathcal{T}_{\mathbf{V}_m} \rightarrow \mathbb{G}$ , which maps  $v'$  in the tangent space  $\mathcal{T}_{\mathbf{V}_m}$  at  $\mathbf{V}_m$  onto the group  $\mathbb{G}$ , is given by  $\mathcal{E}_{\mathbf{V}_m}(\mathbf{V}') = \mathbf{V}_m \exp(\mathbf{V}_m^{-1}\mathbf{V}')$ . The logarithmic map  $\mathcal{L}_{\mathbf{V}_m} : \mathbb{G} \rightarrow \mathcal{T}_{\mathbf{V}_m}$ , meanwhile, is  $\mathcal{L}_{\mathbf{V}_m}(\mathbf{V}) = \mathbf{V}_m \log(\mathbf{V}_m^{-1}\mathbf{V})$ . The matrix exponential and logarithmic operation used here are defined as  $\exp(\mathbf{X}) = \sum_{i=0}^{\infty} \frac{1}{i!} \mathbf{X}^i$  and  $\log(\mathbf{X}) = \sum_{i=1}^{\infty} \frac{(-1)^{i-1}}{i} (\mathbf{X} - \mathbf{I})^i$ .

The space of fundamental matrices -  $\mathbf{F}$ 's, as in Example 3, is the space of those matrices with rank 2. To get a parameterization for this manifold, we employ the singular value decomposition  $\mathbf{F} = \mathbf{U}_1 \mathbf{\Sigma} \mathbf{U}_2^T$ , where  $\mathbf{U}_1$  and  $\mathbf{U}_2$  are both  $3 \times 2$  orthogonal matrices and  $\mathbf{\Sigma}$  is  $2 \times 2$  diagonal positive. It is known that the spaces of all  $3 \times 2$  orthogonal matrices is Stiefel manifold  $\mathfrak{V}_{2,3}$  [14] and thus the spatial alignment manifold  $\mathfrak{M}_S = \mathfrak{V}_{2,3} \times \mathbb{R}^+ \times \mathbb{R}^+ \times \mathfrak{V}_{2,3}$ . For two elements  $\mathbf{V}_1, \mathbf{V}_2$  on  $\mathfrak{V}_{2,3}$ , an intrinsic distance is  $d(\mathbf{V}_1, \mathbf{V}_2) = \sqrt{2 - \text{tr}(\mathbf{V}_1^T \mathbf{V}_2)}$ . The tangent vectors at  $\mathbf{V}_m$ , denoted as  $\mathbf{V}'$ 's, can be represented as  $\mathbf{V}' = \mathbf{V}_m \mathbf{A} + (\mathbf{I} - \mathbf{V}_m \mathbf{V}_m^T) \mathbf{B}$ , where  $\mathbf{A}$  is skew-symmetric and  $\mathbf{B}$  is arbitrary. The exponential map from  $\mathbf{V}'$  to  $\mathbf{V}$ , meanwhile, can be obtained as

$$\mathbf{V} = [\mathbf{V}_m, \mathbf{Q}] \exp \left( \begin{bmatrix} \mathbf{V}_m^T \mathbf{V}' & -\mathbf{R}^T \\ \mathbf{R} & \mathbf{0} \end{bmatrix} \right) \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \tag{3}$$

where  $\mathbf{Q}$  and  $\mathbf{R}$  are the QR-decomposition of  $(\mathbf{I} - \mathbf{V}_m \mathbf{V}_m^T) \mathbf{V}'$ .

### 3.2 The Temporal Alignment Submanifold

As pointed out earlier, in this work we not only account for the temporal misalignment due to synchronization problem and differences in frame rates of the cameras, but also exploit the rate variations within observed dynamic instances of the same category. Rate variation within a fixed time span, *i.e.*,  $[0,1]$ , with global frame rate (scaling) and shift eliminated, is well modeled as a diffeomorphism  $\gamma$  from  $[0,1]$  to  $[0,1]$  with  $\gamma(0) = 0$  and  $\gamma(1) = 1$  [11]. Then, any time warping or misalignment  $W(t)$  under consideration can be written as  $W(t) = k_2\gamma(\frac{t-l_1}{k_1}) + l_2$ , where  $k_1, k_2$  are the positive global scaling factors and  $l_1, l_2$  are the shift factors, defined for  $l_1 \leq t \leq k_1 + l_1$ . Obviously, when we take  $\gamma(t) = t$ ,  $W(t)$  reduces to the temporal affine transformation. Denoting the space of all possible  $\gamma$ 's as  $\mathfrak{d}$ , we can now formally define the temporal alignment submanifold as  $\mathfrak{M}_T = \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R} \times \mathbb{R} \times \mathfrak{d}$ , where  $\mathbb{R}^+ \times \mathbb{R}^+$  accounts for  $k_1, k_2$  and  $\mathbb{R} \times \mathbb{R}$  accounts for  $l_1, l_2$ .

If we let  $\psi = \sqrt{\gamma}$  and the space of all  $\psi$ 's to be  $\ominus$ , then under Fisher-Rao metric (See [15, 16]), the intrinsic distance between  $\psi_1$  and  $\psi_2$  are  $d(\psi_1, \psi_2) = \cos^{-1}(\langle \psi_1, \psi_2 \rangle)$  where  $\langle \psi_1, \psi_2 \rangle = \int_0^1 \psi_1(t)\psi_2(t)dt$ . The exponential map  $\mathcal{E}_{\psi_m} : \mathcal{T}_{\psi_m} \rightarrow \ominus$  for  $\psi' \in \mathcal{T}_{\psi_m}$  is defined as  $\mathcal{E}_{\psi_m}(\psi') = \cos(\langle \psi', \psi' \rangle^{\frac{1}{2}})\psi_m + \frac{\sin(\langle \psi', \psi' \rangle^{\frac{1}{2}})}{\langle \psi', \psi' \rangle^{\frac{1}{2}}}\psi'$ . The logarithmic map  $\mathcal{L}_{\psi_m} : \ominus \rightarrow \mathcal{T}_{\psi_m}$ , which is actually the inverse map of exponential map, is then given by  $\mathcal{L}_{\psi_m}(\psi) = \frac{\arccos(\langle \psi, \psi_m \rangle)}{\langle \psi^*, \psi^* \rangle^{\frac{1}{2}}}\psi^*$ , where  $\psi^* = \psi_m - \langle \psi, \psi_m \rangle \psi$ . Since we have used  $\psi$  instead of  $\gamma$ , the temporal alignment submanifold can also be equivalently represented as  $\mathfrak{M}_T = \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R} \times \mathbb{R} \times \ominus$ .

## 4 Sequential Importance Sampling on the Manifold for Optimal Alignment

It is now clear that the alignment problem (1) becomes an optimization problem on the alignment manifold  $\mathfrak{M}$ . This problem differs from previous works where exhaustive or greedy strategies are employed pertaining to a specific spatio-temporal parameter space, which is usually treated as Euclidean. Meanwhile, the gradient or Newton methods as used previously will tend to fall into local optimum as  $J$  defined on  $\mathfrak{M}$  is normally non-convex and multi-modal. In sum, it is desirable to find an algorithm that accounts for the non-linear manifold of the arguments, converges to the global optimum, and has reasonable computational complexity.

Let us consider the following time-varying state-space model:

$$\begin{bmatrix} \mathbf{p}_{S,h} \\ \mathbf{p}_{T,h} \end{bmatrix} = \begin{bmatrix} \mathcal{E}_{\mathbf{p}_{S,h-1}}(\mathbf{u}_{S,h}) \\ \mathcal{E}_{\mathbf{p}_{T,h-1}}(\mathbf{u}_{T,h}) \end{bmatrix} \tag{4}$$

$$\mathbf{y}_h = J(S^1, S^2, \mathbf{p}_h) - v_h. \tag{5}$$

where  $\mathbf{p}_h = [\mathbf{p}_{S,h}^T, \mathbf{p}_{T,h}^T]^T$  is the parameter state at step  $h$ . We assume that  $\mathbf{p}^*$ , the optimal alignment, is not directly observable, while at step  $t$  we observe  $\mathbf{y}_t$ . Moreover, we let  $\mathbf{u}_{S,h} \sim \mathcal{N}(\mathbf{0}, (\sigma_S/h)^2 \mathbf{I})$ ,  $\mathbf{u}_{T,h} \sim \mathcal{N}(\mathbf{0}, (\sigma_T/h)^2 \mathbf{I})$ , where  $\sigma_S^2$  and  $\sigma_T^2$  are both small numbers. By construction (details below) we may let  $v_h$  to be a non-negative random variable with an appropriate density function (e.g., exponential  $\mathcal{E}(\lambda)$  in this work). Equivalently, we may represent the state transition and observation model as  $p(\mathbf{p}_{S,h} | \mathbf{p}_{S,h-1}) \sim \exp(-\frac{d^2(\mathbf{p}_{S,h}, \mathbf{p}_{S,h-1})}{2(\sigma_T/h)^2})$ ,  $p(\mathbf{p}_{T,h} | \mathbf{p}_{T,h-1}) \sim \exp(-\frac{d^2(\mathbf{p}_{T,h}, \mathbf{p}_{T,h-1})}{2(\sigma_T/h)^2})$ , where  $p(\mathbf{p}_h | \mathbf{p}_{h-1}) \sim p(\mathbf{p}_{S,h} | \mathbf{p}_{S,h-1})p(\mathbf{p}_{T,h} | \mathbf{p}_{T,h-1})$ , and  $p(\mathbf{y}_h | \mathbf{p}_h) \sim \exp(\lambda(\mathbf{y}_h - J(S^1, S^2, \mathbf{p}_h)))$ .

The motivation as to why we formulate a state space model is to be able to recursively compute the Maximum A Posteriori (MAP) estimate of the parameter state  $p(\mathbf{p}_h | \mathbf{y}_h, \mathbf{y}_{h-1}, \dots, \mathbf{y}_0)$ . From the recursion  $p(\mathbf{p}_h | \mathbf{y}_h, \mathbf{y}_{h-1}, \dots, \mathbf{y}_0) \propto p(\mathbf{y}_h | \mathbf{p}_h) \int p(\mathbf{p}_h | \mathbf{p}_{h-1})p(\mathbf{p}_{h-1} | \mathbf{y}_{h-1}, \mathbf{y}_{h-2}, \dots, \mathbf{y}_0)d\mathbf{p}_{h-1}$ , we know that the posterior probability of the alignment  $p(\mathbf{p}_h | \mathbf{y}_h, \mathbf{y}_{h-1}, \dots, \mathbf{y}_0)$  is equal to the posterior probability at the previous step  $p(\mathbf{p}_{h-1} | \mathbf{y}_{h-1}, \mathbf{y}_{h-2}, \dots, \mathbf{y}_0)$  smoothed by the state transition probability  $p(\mathbf{p}_h | \mathbf{p}_{h-1})$  and weighted by the likelihood  $p(\mathbf{y}_h | \mathbf{p}_h)$ . Therefore, by constructing a decreasing sequence  $\{\mathbf{y}_h\}_{h=0,1,\dots}$  and letting  $\sigma_S, \sigma_T$  be small,  $p(\mathbf{p}_h | \mathbf{y}_h, \mathbf{y}_{h-1}, \dots, \mathbf{y}_0)$  is expected to be continuously increasing and peaking at the the optimal alignment  $\mathbf{p}^*$ . In other words, the MAP estimate of the parameter state will give the optimal alignment.

The above Bayesian recursive estimation is realized in a Monte Carlo manner. In particular, the construction of appropriate observation sequence  $\{\mathbf{y}_h\}_{h=0,1,\dots}$  come up naturally from the Monte Carlo samples. We propose the SIS algorithm on the alignment manifold as follows. Note that the proposed algorithm handles states evolving on the Riemannian manifold rather than the conventional Euclidean space, thus is different from most existing particle filters and their variations. Bayesian recursive filtering using particles has been proposed for specific manifolds in the context of tracking [17–20], while the following approach is generally applicable for various alignment manifolds. Furthermore, we formulate the static optimization problem into a dynamic state space model, which provides insight on applications of SIS to new problems beyond tracking.

**Algorithm.** *SIS on the alignment manifold.*

1) Initialization. Specify an initial distribution  $p_0$  defined on  $\mathfrak{M}$  and draw i.i.d. samples  $\{\mathbf{p}_0^k\}_{k=1}^K$  from  $p_0$ . Let  $h = 1$ .

2) Importance Sampling. Sample  $\hat{\mathbf{p}}_h^k$  from  $p(\mathbf{p}_h^k | \mathbf{p}_{h-1}^k)$ . For this purpose, generate  $\mathbf{u}_{S,h}^k$  from  $\mathcal{N}(\mathbf{0}, (\sigma_S/h)^2 \mathbf{I})$  and  $\mathbf{u}_{T,h}^k$  from  $\mathcal{N}(\mathbf{0}, (\sigma_T/h)^2 \mathbf{I})$ . Then apply exponential maps  $\hat{\mathbf{p}}_{S,h}^k = \mathcal{E}_{\mathbf{p}_{S,h-1}^k}(\mathbf{u}_{S,h}^k)$  and  $\hat{\mathbf{p}}_{T,h}^k = \mathcal{E}_{\mathbf{p}_{T,h-1}^k}(\mathbf{u}_{T,h}^k)$ .

3) Constructing observation. Let

$$\mathbf{y}_h = \min_k J(S^1, S^2, \hat{\mathbf{p}}_h^k). \quad (6)$$

If  $\mathbf{y}_h > \mathbf{y}_{h-1}$ ,  $\mathbf{y}_h \leftarrow \mathbf{y}_{h-1}$ .

4) Weighting. Approximate the new posterior probability by

$$q_h(\mathbf{p}_h) = \sum_{k=1}^K w_h^k \delta(\mathbf{p}_h - \hat{\mathbf{p}}_h^k), \quad (7)$$

where  $\delta$  is the Kronecker delta,  $w_h^k \propto p(\mathbf{y}_h | \hat{\mathbf{p}}_h^k)$  and  $\sum_{k=1}^K w_h^k = 1$ .

5) Importance resampling. Draw i.i.d. samples  $\{\mathbf{p}_h^k\}_{k=1}^K$  from  $q_h(\mathbf{p}_h)$ .

6) Stop if a stopping criteria is satisfied; Otherwise,  $h \leftarrow h + 1$  and go to 2).

Step 3) follows from the observation equation in the proposed state-space model, and this construction of observation  $\mathbf{y}_h$  plays an important role in the above algorithm. By letting  $\mathbf{y}_h$  to be the minimum value of the alignment cost function, Monte Carlo samples that lead to a lower cost will receive higher importance weights when applying the weighting step. Consequently, the Monte Carlo samples (particles) will tend to concentrate around the minima of the alignment cost function, including the global minimum. With a proper initialization of samples over  $\mathfrak{M}$ , the optimal  $\mathbf{p}^*$  will be located more and more accurately during the coarse-to-fine particle propagation. The operation  $\mathbf{y}_h \leftarrow \mathbf{y}_{h-1}$  when  $\mathbf{y}_h > \mathbf{y}_{h-1}$  guarantees non-increasing  $\mathbf{y}_h$ .

The initialization of the particles is case dependant. As an example for the spatial alignment submanifold of  $\mathbb{A}(2)$ , we may generate independent, uniformly distributed samples over the corresponding Lie algebra  $\mathfrak{D}(2)$  and exponentially map them onto  $\mathbb{A}(2)$ . For the temporal alignment submanifold, we may also generate uniform distributed samples over the tangent space at  $\gamma(t) = t$  together with uniform samples from  $\mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R} \times \mathbb{R}$ . The stopping criteria, meanwhile, can be flexible as well.  $0 < \mathbf{y}_{h-1} - \mathbf{y}_h < \epsilon$  is a useful one. The final MAP estimate of  $\mathbf{p}^*$ , can be simply taken as  $\hat{\mathbf{p}}^* = \arg \min_k J(S^1, S^2, \hat{\mathbf{p}}_h^k)$  after the algorithm stops at step  $h$ .

## 5 Empirical Evaluation

We have applied the algorithm described above to three different datasets for the same purpose of spatial-temporal alignment, while these datasets represent different spatio-temporal signals originated from videos. Specifically, we looked into the alignment of point trajectories, deforming shape sequences, as well as videos themselves. The alignment objectives and alignment manifolds corresponding to each datasets vary, while the SIS procedure is the same for all. In each experiment, we select appropriate state-of-art methods or design baseline(s) for comparison, while the purpose of these comparisons is simply to show how the inclusion of temporal warping submanifold, formulation of the aligning procedure as a recursive estimation of the state-space model, and the Monte Carlo approach help advance the state-of-art performance on practical data.

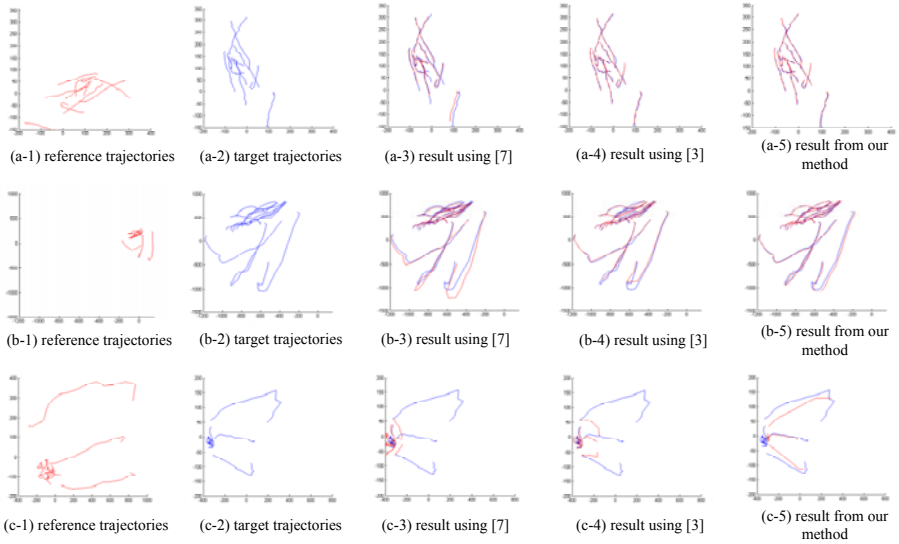
### 5.1 Evaluation with Point Trajectories

We first evaluate our method with point trajectories, which are essentially the input to feature-based methods. In this paper we make use of the GaTech Football

Multi-Trajectory Dataset [21, 22]. This dataset contains 55 sets of trajectories and in each set there are eleven trajectories corresponding to the movements of the eleven offensive players in a play. The sets are organized into categories, each of which contains all realizations of the same play strategy (specified by the playbook). In other words, trajectory sets in the same category are samples of the same ‘activity’, thus resembling each other (on the ground plane) though intra-category variations exist. However, they are observed in different viewpoint and executed at different and varying rates. In each set the roles of players are annotated and thus the trajectory correspondence between two sets is available to us.

We model the spatial misalignments to be a planar homography and thus the spatial alignment submanifold becomes  $\mathbb{S}\mathbb{L}(3)$ . The misalignment cost  $J$  is simply taken as the average distances between point pairs from all trajectory pairs across the whole time span. We perform two types of experiments, in the first of which we select a set of trajectories and transform it with a typical view change (homography) and a specific time warping to get the other, and then we align the two. We do so on all 55 sets. In the second type, we randomly select a total of 40 pairs of sets, each pair being the samples of the same play type (activity), and then we align these pairs. For comparative purposes, we implemented two state-of-art methods [3, 7] that address similar task as ours. The approach in [7] assumes affine temporal misalignment only, and the strategy in [3] uses Dynamic Time Warping (DTW) to determine the non-linear temporal misalignment. The preprocessing modules of tracking and correspondence in the two methods are unnecessary as the dataset has provided trajectory and correspondence information, and thus a common basis is shared among all implementations for comparison. Note that [7] mainly focus on temporal alignment, and to add spatial alignment into it we simply estimate a planar homography with the points from the temporally aligned trajectories. Meanwhile, when using [3] we take alternations between DTW and gradient-descent-based homography estimation (on all corresponding points collected from all temporally aligned frames) to get the final alignment parameters. (Note that though DTW is globally optimal in 1-D temporal dimension, when placed into alternations between spatial and temporal submanifolds the combined search may not necessarily be so, and thus the alternating process is a greedy search.) For our method, we get the initial particles by generating random samples in the tangent space at the homography estimated from the first pair frames and in the tangent space at  $\gamma(t) = t$ .

Samples of the results are shown in Figure 1, where in the first two columns are the two trajectory sets to be aligned toward each other, and the following two columns show alignment results using the state-of-art methods and our method. Each of the three rows, meanwhile, represents a typical experimental setting: in row (a) the target is a generated misaligned version of the reference, in row (b) the reference and target are a real pair with similar realization but undergoing significant misalignment, and in row (c) the two are a real pair with significant variation from each other.



**Fig. 1.** Samples of the alignment results on point trajectories

To quantitatively understand the performance of the alignment methods, we recorded the average distance of point pairs from aligned trajectory pairs, and show the results in Table 1. Note that the statistics is from the 40 real pairs rather than the generated ones.

**Table 1.** Average residual misalignments between the aligned trajectory pairs

	mean	standard deviation
Using [7]	15.9	8.6
Using [3]	13.1	6.8
Our method	10.0	3.6

## 5.2 Evaluation with Deforming Shape Sequences

Sequences of deforming shapes are typical mid-level features extracted from original videos containing the deforming objects of interest. In this experiment we use silhouette sequences from the USF Gait Database [23] to demonstrate the performance of our method. We randomly select 20 sequence pairs, each with the same shoe types, carrying conditions, surface types, and walking directions, but observed at two different times. For efficiency, in each sequence we only consider the segment of frames of the first two walking circles. The spatial misalignment within each pair is modeled as affine and is actually less significant compared to the GaTech Football Multi-Trajectory Dataset, and the main focus is on the effect of taking non-linear rate warping into account in addition to linear scaling and shift. For comparison, we implemented the gradient descent algorithm



presented in [5] and designed one more baseline. The designed baseline alternates between DTW (on all frames from spatial alignment) and gradient-descent-based affine estimation (on all frame pairs temporally aligned), and thus is a greedy search. The cost function is simply taken as the sum of pixel-wise absolute differences. For [5], the initial spatial parameter is estimated as a translation between the leading frames and the initial temporal parameter is taken as  $\gamma(t) = t$ . For our method, Monte Carlo samples are generated from Gaussians in the tangent spaces at the initial parameters.

We show two sample results in Figure 2, where each of the five rows for each sequence pair is explained in the caption of the figure. The average residual misalignment errors (in pixels) for all 20 pairs are shown in Table 2. Note that all three methods perform well due to mild spatial misalignment and near-affine temporal misalignment, while our method achieves improvement over [5] by allowing non-linear warping effect, and the improvement over alternating DTW and affine estimation should be credited to better global convergency.

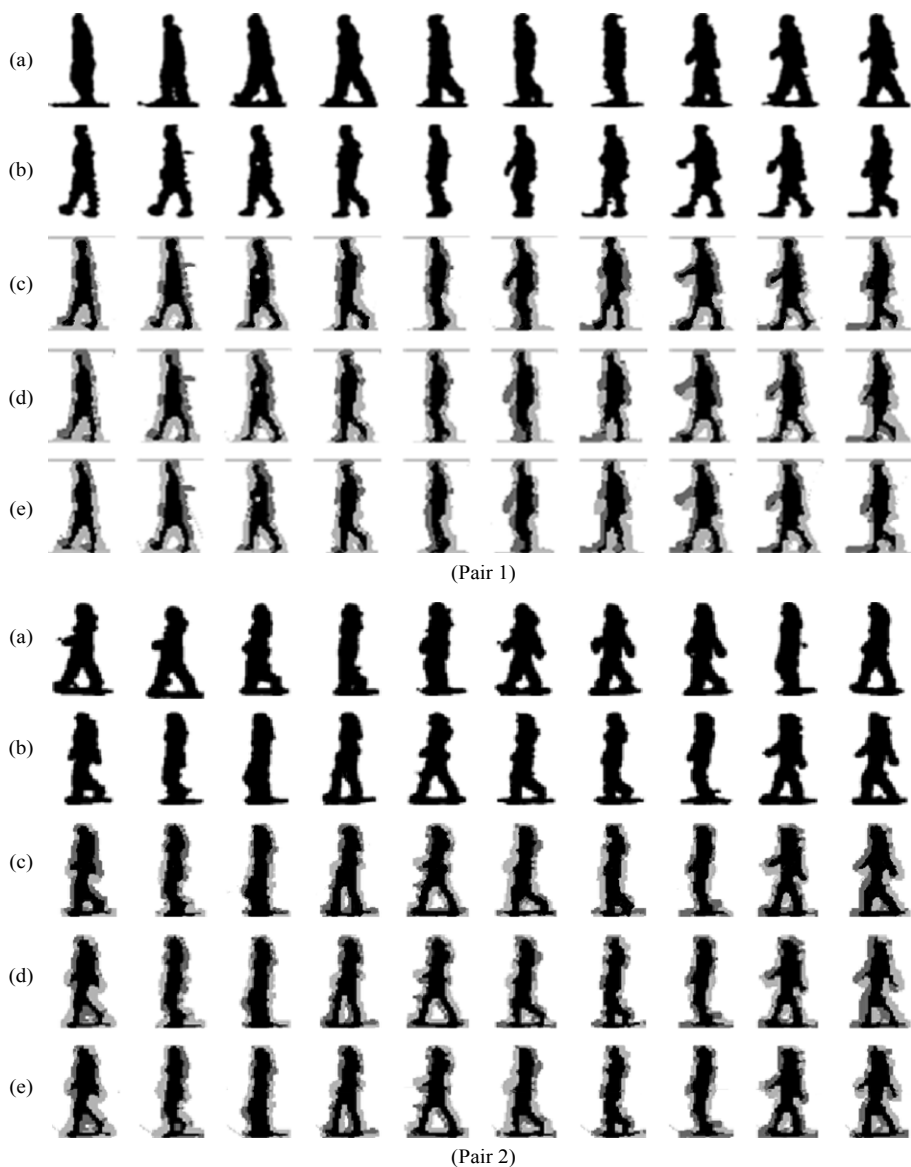
**Table 2.** Average residual misalignments between the pairs of shape sequences

	mean	standard deviation
Using [5]	23.5	6.4
Alternating DTW and affine estimation	26.7	6.8
Our method	21.3	7.1

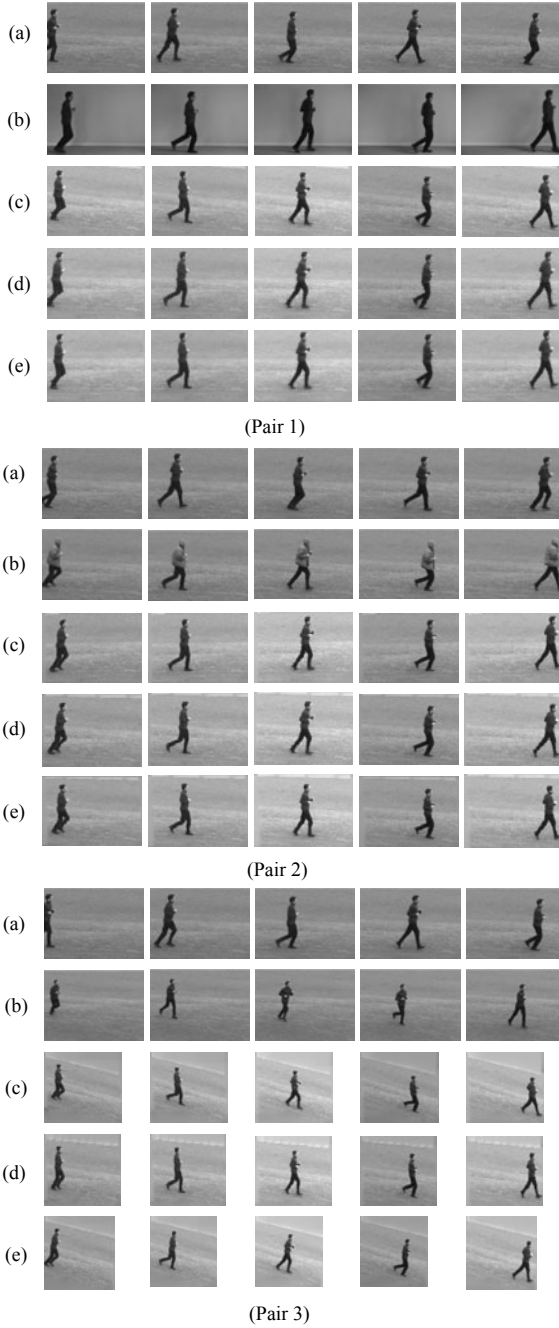
### 5.3 Evaluation with Human Action Videos

In the third set of experiments, we work with human action videos directly. We use the KTH database [24], in which the semantically meaningful signal is human motion. We randomly select 30 pairs of sequences, each pair performing the same action, but moderate variations in clothing, background, or view angle exist within the pair. For efficiency, again for each sequence we only keep a segment of frames including human motion but discard pure background frames. The spatial misalignment within each pair is affine [10], and the misalignment cost is the spatio-temporal correlation used by [10] but on optical flow extracted from consecutive frames. For comparison, we implemented [10] and the method that alternates between DTW and affine estimation as in previous section. The initial spatial parameter, when necessary, is estimated as the translation between the leading frames and the initial temporal parameter is taken as  $\gamma(t) = t$ . Meanwhile, Monte Carlo samples are generated from Gaussians in the tangent spaces at the initial parameters too.

We show three sample results in Figure 3, where each of the five rows for each sequence pair has the same interpretation as in the previous section. Substantial execution rate variations exist within every pair, and changes in clothing, background, or view angle also exist. There is not a numerical criterion to evaluate the performance on aligning real videos, and by qualitative observation the proposed method performs comparatively well as the baselines, and is visually more close to the target when undergoing a larger view change (pair 3).



**Fig. 2.** Samples of the alignment results on deforming shape sequences from USF Gait Database. For each pair, (a) is the reference sequence and (b) is the target. (c), (d), and (e) give the alignment results (transformed sequence overlaid onto target) using our method, the method in [5], and the method that alternates between DTW and spatial alignment. The black, dark shaded, light shaded, and white areas denote true positive, false negative, false positive and true negative respectively. In other words, the black and dark shaded areas constitute the silhouette of the target, while the black and lighted shaded areas constitute the transformed silhouette. Therefore, a larger black area implies a better alignment.



**Fig. 3.** Samples of the alignment results on KTH dataset. For each pair, (a) is the reference sequence and (b) is the target. (c), (d), and (e) give the alignment results (transformed sequence overlaid onto target) using our method, the method in [10], and the method of alternation between DTW and spatial alignment.

As is true for many efforts involving particle filters, the proposed method is computationally more demanding than greedy search, but much less expensive than exhaustive approaches. This trade-off, however, leads to improved performance as demonstrated in the previous subsections. The time complexity depends on the number of particles used. The convergence, on the other hand, turns out to be fast. In this section all results are obtained with 1000 particles and less than twenty iterations. Another issue is that  $\mathfrak{d}$  is by definition infinitely dimensional, while in all experiments we approximated the  $\gamma$ 's with non-decreasing sequences valued from 0 to 1 of length 20.

## 6 Discussion

This work assumes that the parametric manifolds are known a priori; alignment problems without knowing the specific form of the manifolds deserve exploration as well. It is also desirable to remove the assumption regarding relative stationarity between cameras. Though we pursue global optimum in the algorithm and empirically observe improved solution, we have not theoretically proved any properties regarding asymptotic convergence. A theoretical study on geometrical SIS method will be important. We will also look into other efficient search schemes like stochastic gradient descent. By generalizing the considered manifolds and cost functions, we will extend the proposed strategy of stochastic optimization on geometric spaces for other problems (*e.g.* face alignment on Grassmann manifold [25]). We hope this can bring new insights and improved performance to a larger number of vision applications.

**Acknowledgement.** The authors thank the anonymous reviewers for valuable comments and suggestions. This work was supported by the ONR Grant N00014-09-1-0664.

## References

1. Lee, L., Romano, R., Stein, G.: Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 758–767 (2000)
2. Wolf, L., Zomet, A.: Sequence to sequence self calibration. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*. LNCS, vol. 2351, pp. 370–382. Springer, Heidelberg (2002)
3. Rao, C., Gritaiand, A., Shah, M., Syeda-Mahmood, T.: View-invariant alignment and matching of video sequences. In: *ICCV (2003)*
4. Laptev, I., Belongie, S., Perez, P., Wills, J.: Periodic motion detection and segmentation via approximate sequence alignment. In: *ICCV (2005)*
5. Caspi, Y., Simakov, D., Irani, M.: Feature-based sequence-to-sequence matching. *International Journal of Computer Vision* 68, 53–64 (2006)
6. Wolf, L., Zomet, A.: Wide baseline matching between unsynchronized video sequences. *International Journal of Computer Vision* 68, 43–52 (2006)
7. Padua, F., Carceroni, R., Santos, G., Kutulakos, K.: Linear sequence-to-sequence alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 304–320 (2010)

8. Caspi, Y., Irani, M.: A step towards sequence-to-sequence alignment. In: CVPR (2000)
9. Caspi, Y., Irani, M.: Alignment of non-overlapping sequences. In: ICCV (2001)
10. Ukrainitz, Y., Irani, M.: Aligning sequences and actions by maximizing space-time correlations. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 538–550. Springer, Heidelberg (2006)
11. Veeraraghavan, A., Srivastava, A., Roy-Chowdhury, A., Chellappa, R.: Rate-invariant recognition of humans and their activities. *IEEE Transactions on Image Processing* 18(6), 1326–1339 (2009)
12. Zhou, F., de la Torre, F.: Canonical time warping for alignment of human behavior. In: NIPS (2009)
13. Gordon, N., Salmond, D., Smith, A.: Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F on RADar and Signal Processing* 140, 107–113 (1993)
14. Arias, T., Edelman, A., Smith, S.: The geometry of algorithms with orthogonality constraints. *SIAM Journal of Matrix Analysis and Applications* 20, 303–353 (1998)
15. Maybank, S.: The Fisher-Rao metric for projective transformations of the line. *International Journal of Computer Vision* 63, 191–206 (2005)
16. Srivastava, A., Jermyn, I., Joshi, S.: Riemannian analysis of probability density functions with applications in vision. In: CVPR (2007)
17. Srivastava, A., Klassen, E.: Bayesian and geometric subspace tracking. *Advances in Applied Probability* 36, 43–56 (2004)
18. Wu, Y., Wu, B., Liu, J., Lu, H.: Probabilistic tracking on riemannian manifolds. In: ICPR (2008)
19. Kwon, J., Lee, K.M., Park, F.C.: Visual tracking via geometric particle filtering on the affine group with optimal importance functions. In: CVPR (2009)
20. Porikli, F., Pan, P.: Refressed importance sampling on manifolds for efficient object tracking. In: AVSS (2009)
21. Li, R., Chellappa, R., Zhou, S.K.: Learning multi-modal densities on discriminative temporal interaction manifold for group activity recognition. In: CVPR (2009)
22. Li, R., Chellappa, R.: Group motion segmentation using a spatio-temporal driving force model. In: CVPR (2010)
23. Sarkar, S., Phillips, P.J., Liu, Z., Robledo, I., Grother, P., Bowyer, K.W.: The human id gait challenge problem: Data sets, performance, and analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 162–177 (2005)
24. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local svm approach. In: ICPR (2004)
25. Lui, Y.M., Beveridge, J.R.: Grassmann registration manifolds for face recognition. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 44–57. Springer, Heidelberg (2008)

# Supervised Label Transfer for Semantic Segmentation of Street Scenes

Honghui Zhang<sup>1</sup>, Jianxiong Xiao<sup>2</sup>, and Long Quan<sup>1</sup>

<sup>1</sup> The Hong Kong University of Science and Technology  
{honghui, quan}@cse.ust.hk

<sup>2</sup> Massachusetts Institute of Technology  
jxiao@csail.mit.edu

**Abstract.** In this paper, we propose a robust supervised label transfer method for the semantic segmentation of street scenes. Given an input image of street scene, we first find multiple image sets from the training database consisting of images with annotation, each of which can cover all semantic categories in the input image. Then, we establish dense correspondence between the input image and each found image sets with a proposed KNN-MRF matching scheme. It is followed by a matching correspondences classification that tries to reduce the number of semantically incorrect correspondences with trained matching correspondences classification models for different categories. With those matching correspondences classified as semantically correct correspondences, we infer the confidence values of each super pixel belonging to different semantic categories, and integrate them and spatial smoothness constraint in a markov random field to segment the input image. Experiments on three datasets show our method outperforms the traditional learning based methods and the previous nonparametric label transfer method, for the semantic segmentation of street scenes.

## 1 Introduction

Semantic segmentation of street scenes is an important and interesting researching topic for scene understanding [1, 2] and image based modeling in cities and urban areas [3–6]. Traditional methods to solve this problem, such as [7–11], typically work with a fixed-number of object categories and train generative or discriminative models for each category. Recently, with the increasing availability of image collections with annotation, large database-driven approaches have shown the potential for nonparametric methods in several applications, such as object and scene recognition [12] and semantic segmentation [13]. Instead of training sophisticated parametric models, these methods try to reduce the inference problem for an unknown image to the problem of matching it to an existing set of annotated images by exploiting local similarity between images, which is addressed as label transfer in [13].

With scenes limited to street scenes, semantic segmentation is a suitable candidate for the application of the label transfer. Ideally, for an testing image, if

some high quality matches are contained in the training database with annotation, and we have a proper way to find them, the label transfer method [13] can work very well. However, in most cases, high quality matches are hardly available [14], even in street scenes, with exponential number of different object combinations within each scene. In addition, most existing methods for searching similar images are based on holistic similarity between images, such as widely used GIST descriptor [15]. They are computed on the layout of global content in images, which does not care about the quality of local matches. Without guarantee of the local similarity between images, establishing semantically correct correspondence between images become very challenging. On the other hand, traditional methods [7-11] focus on local similarity for classifier training, and doesn't care about holistic image-level suitability.

This motivates us to investigate whether the combination of the traditional learning based methods and the pure label transfer can improve the performance of semantic segmentation of street scenes. In this paper, we propose a supervised label transfer method for semantic segmentation of street scenes, which introduces supervised classification models into the pure label transfer, to classify obtained matching correspondences and reject those untrusted matching correspondences.

The paper is structured as follows: a brief overview of our method is given in Section 2. In Section 3 and 4, the proposed KNN-MRF matching scheme and the supervised classification models for label transfer are introduced respectively. Then, the confidence inference, generation of segmentation mask for the input image and how to retrieve proper source for the supervised label transfer method are explained in Section 5. Finally, we evaluate and compare our method with related works in Section 6, and conclude in Section 7.

## 2 Overview

For a given input image, our method starts from finding some proper image sets with annotation from an existing database, each of which contain multiple images with annotation that cover all semantic categories in the input image. As mentioned above, it is difficult to find a single overall good match for query images. Some parts of the query image may be matched well, while some other parts could be totally missed. Based on this observation, it is claimed that a query image should be explained by a spatial composite of different regions taken from different images [14]. Inspired by this idea, we propose a new matching scheme for the label transfer that matches the given input image to each of the retrieved image sets, instead of matching it to a single image like [13].

To be specific, we perform a matching scheme that we call KNN-MRF matching between the input image and each of the retrieved image sets to establish a dense correspondence on super-pixel level. Then, it is followed by a matching correspondences classification step that uses some trained classification models to classify the matching correspondences and discard correspondences that are classified as semantically incorrect correspondences. Finally, with those matching correspondences that are classified as semantically correct correspondences,

we infer the confidence value of each basic matching element belonging to different categories, and then integrate these inferred confidence cues and spatial smoothness constraint into a markov random field to segment the input image. An outline of our method is given in Fig. 1. In the following section, we will first introduce the KNN-MRF matching scheme for an input image and an image set.

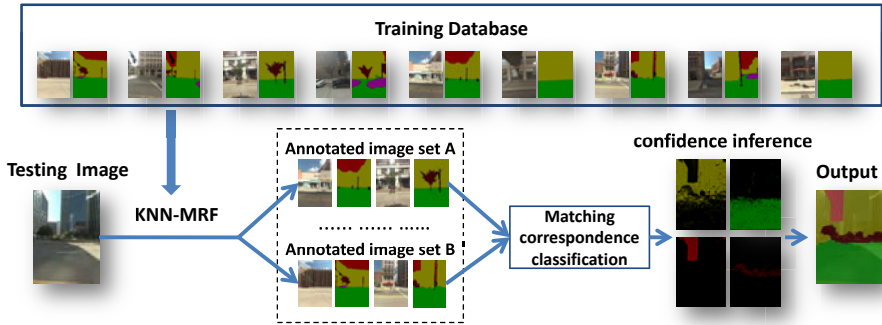


Fig. 1. Outline of our supervised label transfer algorithm

### 3 KNN-MRF Matching

In order to transfer labels of annotated images to an input image, we need to establish dense semantically correct correspondence between the input image and the annotated images. To make the matching process efficient, we use super pixel as basic matching element, instead of pixel as [13] did. As most super pixels are semantically coherent, using super pixel as basic matching element would be proper. For each image, a graph  $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$  is defined, with each vertex  $p \in \mathcal{V}$  in the graph denotes one super pixel in the image, while the edges  $\mathcal{E}$  denotes the neighboring relationship between super pixels. Then we could formulate the matching problem as a simplified inexact graph matching problem [16]. More formally, given two graphs  $\mathcal{G}_I = (\mathcal{V}_I, \mathcal{E}_I)$  and  $\mathcal{G}_A = (\mathcal{V}_A, \mathcal{E}_A)$  that denote the input image and an retrieved image set for it respectively, the inexact graph matching problem consists in searching for a mapping from  $\mathcal{V}_I \rightarrow \mathcal{V}_A$ , with a constraint that each node in  $\mathcal{V}_I$  will be matched to another node in  $\mathcal{V}_A$ . For the retrieved image set, we simply combine the graph for each image of it together to form a larger graph.

To solve this graph matching problem, we propose an efficient matching scheme that we call KNN-MRF matching scheme, named by the way we solve the problem. Given an input image  $I$  and an image set  $\{A_i\}_{i=1}^N$  with annotation, we first find the  $K$ -Nearest-Neighbor(KNN) for each super pixel in  $I$  from  $\{A_i\}_{i=1}^N$  (in our implementation  $K = 5$ ). Then we use a Markov random field built upon the graph  $\mathcal{G}$  defined for  $I$  with the following energy function:

$$E(C) = \sum_{p_i \in \mathcal{V}} S(C_i) + \alpha \sum_{e_{ij} \in \mathcal{E}} f(C_i, C_j) \quad (1)$$



The candidate label set  $\{C_i\}_{i=1}^K$  for each super pixel in  $I$  consists of the index set  $\{1, 2, \dots, K\}$  for the corresponding  $K$  nearest neighbor from  $\{A_i\}_{i=1}^N$ , and  $\mathcal{E}$  contains all the spatial neighborhood. In the energy function,  $S(C_i)$  denotes the matching distance of each node to its  $C_i$ -th nearest neighbor.  $f(C_i, C_j) = 0$  if the  $C_i$ -th nearest neighbor and  $C_j$ -th nearest neighbor of two neighboring super pixels in  $I$  are also neighboring, else  $f(C_i, C_j) = 1$ . For neighboring super pixels in  $I$ , by setting the smooth term with this way, matching to neighboring nearest neighbor will be given more preference. The alpha expansion algorithm [17] can be used to optimize the energy function and obtain a dense matching configuration. An example with detailed explanation is given in Fig. 2.

### 3.1 Superpixel Descriptor

Visual word has already been proven to be powerful in many visual problems, like object recognition and segmentation. We will combine it with super pixel to describe an image. For each image, it is first decomposed into many coherent regions, super pixels, by using the turbo pixel algorithm [18] which could make the size of super pixels balanced. Then the visual word descriptor for each super pixel is generated by quantizing features of pixels contained in the super pixel with a hierarchical k-mean clustering tree. In our implementation, we use the Texton feature [9] of pixels. To reduce time cost of searching the  $K$  nearest neighbor, for each image in the database, the visual word descriptor for each super pixel in it is precomputed and organized in a KD-tree for later reference.

### 3.2 Distance Metric

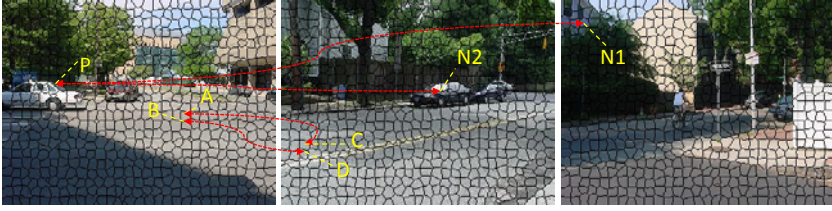
For the KNN-MRF matching between  $I$  and  $\{A_i\}_{i=1}^N$ , the distance metric used to retrieve the  $K$ -Nearest-Neighbor is defined as:

$$D(p, q) = \|(D_p - D_q)\|^2 + \beta(1 - [L_q \in R]) \quad (2)$$

where  $p$  and  $q$  are two super pixels in  $I$  and  $\{A_i\}_{i=1}^N$  respectively,  $D_p$  and  $D_q$  are the feature descriptor of them.  $L_q$  is the label of  $q$  and known from the annotation of  $\{A_i\}_{i=1}^N$ , and  $R$  is the image level prior of the semantic categories contained in  $I$ . The image level prior is incorporated into the distance metric, and it has already been shown in [8], image level prior can improve semantic segmentation performance. When no image level prior is available, we set  $\beta = 0$ .

## 4 Matching Correspondences Classification

For the super-pixel matching correspondences obtained by the KNN-MRF matching between the input image  $I$  and the image set  $\{A_i\}_{i=1}^N$  with annotation, we denote them as  $\{\langle T_j, D_{T_j}, L_{T_j}, S_j, D_{S_j}, L_{S_j} \rangle\}$ .  $T_j$  and  $S_j$  are the super pixels matched together in  $I$  and  $\{A_i\}_{i=1}^N$  respectively,  $D_{T_j}$  and  $D_{S_j}$  are the descriptor for them, and  $L_{T_j}$  and  $L_{S_j}$  are the labels of them. To reduce the number



**Fig. 2.** KNN-MRF Matching between an input image (a) and an image set that consists of (b) and (c): for each super pixel  $P$  in (a), candidate label set for  $P$  in the energy function (II) consists of the  $K$  nearest neighbor of  $P$ ,  $\{N_i\}$  in (b) and (c); for neighboring super pixels  $A$  and  $B$ , matching to neighboring super pixel  $C$  and  $D$  will be given more preference by setting the smoothness term of the energy function (II) properly

of mismatch( $L_{T_j} \neq L_{S_j}$ ), we train some matching correspondences classification models with the extremely randomized forest [19] to classify the obtained matching correspondences and discard those semantically incorrect matching correspondences, or mismatches.

For each matching correspondence  $\langle T_j, D_{T_j}, L_{T_j}, S_j, D_{S_j}, L_{S_j} \rangle$ ,  $L_{S_j}$  is known from the annotation associated with  $\{A_i\}_{i=1}^N$ , so to distinguish whether it is a mismatch, we can reduce the problem to distinguish whether it is a mismatch for the certain category  $L_{S_j}$ . Therefore, instead of training a general classification model for all matching correspondences, we train a unary matching correspondences classification model for each category. The main advantage of training multiple matching correspondences classification models is improved performance, since certain cues and features are important for some categories and not for others.

To generate training samples for the matching correspondences classification model of a certain category  $L$ , we randomly select some image pairs  $\{\langle A_m, A_n \rangle\}$  with annotation from the database, with both  $A_m$  and  $A_n$  containing  $L$ . Then for each super pixel in  $A_m$ , we find a nearest neighbor in  $A_n$ . By doing this, we can obtain many matching correspondences  $\{\langle T_k, D_{T_k}, L_{T_k}, S_k, D_{S_k}, L_{S_k} \rangle\}$ , where  $L_{T_k}$  and  $L_{S_k}$  have already been known. For a matching correspondence  $\langle T_k, D_{T_k}, L_{T_k}, S_k, D_{S_k}, L_{S_k} \rangle$ , it is taken as a positive training sample, if  $L_{T_k} = L_{S_k} = L$ , and a negative training sample if  $L_{T_k} \neq L_{S_k}$ ,  $L_{T_k} = L$  or  $L_{T_k} \neq L_{S_k}$ ,  $L_{S_k} = L$ . In detail, given a correspondence  $\langle T_k, D_{T_k}, L_{T_k}, S_k, D_{S_k}, L_{S_k} \rangle$ , an appearance difference vector

$$V = |D_{T_k} - D_{S_k}|$$

combined with a position feature, the offset of their centers normalized with respect to the image width and height respectively.

$$\text{offset} = (|X_{T_k} - X_{S_k}|, |Y_{T_k} - Y_{S_k}|)$$

$\langle V, \text{offset} \rangle$  is used as the feature vector for the training samples of the matching correspondences classification model of category  $L$ .

The testing of a matching correspondence  $\langle T, D_T, L_T, S, D_S, L_S \rangle$  is similar. Extract the feature vector  $\langle V, \text{offset} \rangle$  first, then test it with the trained matching correspondences classification model for category  $L_S$ . If it is classified as a mismatch, we discard this matching correspondence.

## 5 Confidence Inference and MRF Integration

### 5.1 Selection of Proper Image Sets for Label Transfer

Given an input image, the first thing we need to do is selecting some proper image sets from the database, which can cover all semantic categories in the input image. We use the following way to find these image sets. First, we predict the image level prior  $R$  of the input image by retrieving the top  $K$ -nearest neighbor from the database with GIST matching [15], which has been proven a good way to predict the contents of images in [20]. Then we extract a subset  $V$  from the database: for each category  $L \in R$ , we retrieve the top  $K$ -nearest neighbor from the database with GIST matching and put them in  $V$ , with a constraint that they should also contain the category  $L$ . With this subset  $V$ , to generate an image set for matching, we randomly select some images from  $V$  until all categories in  $R$  have already been covered in at least one of the selected images. By repeating this process, we can get multiple image sets for the KNN-MRF matching.

### 5.2 Confidence Inference

With the multiple image sets obtained, we perform the KNN-MRF matching between the input image and each image set, followed by the matching correspondences classification. With matching correspondences  $\{\langle T, D_T, L_T, S_n, D_{S_n}, L_{S_n} \rangle\}$  for each super pixel  $T$  in the input image, the confidence of  $T$  belonging to different categories are estimated as:

$$p(L_T = l|T) = \frac{F_T(l)W(l)}{\sum_{j=1}^L F_T(j)W(j)}, l = 1, 2, 3 \dots K \quad (3)$$

where  $F_T(l)$  is the occurrence of  $\{\langle T, D_T, L_T, S_n, D_{S_n}, L_{S_n} : L_{S_n} = l \rangle\}$ . In real situation, the frequency of occurrence of different category  $\{P_k, k = 1, 2, \dots, K\}$  could be quite different, which could make the matching bias toward categories with high frequency of occurrence. To overcome this problem, we introduce the weight term  $W(l) = 1/P_k$  to balance the contribution of different categories.

### 5.3 Influence of Matching Correspondences Classification

In this part, we will analysis the influence of matching correspondences classification for the confidence inference. Suppose no matching correspondences

classification is done after the KNN-MRF matching, then the confidence of  $T$  belonging to different categories would be estimated as:

$$p'(L_T = l|T) = \frac{F'_T(l)W(l)}{\sum_{j=1}^L F'_T(j)W(j)}, l = 1, 2, 3...K \tag{4}$$

where  $F'_T(l)$  is the occurrence of  $\{(T, D_T, L_T, S_n, D_{S_n}, L_{S_n} : L_{S_n} = l)\}$  obtained by KNN-MRF matching. Suppose the true label of  $T$  is  $i$  and the classification precision of each matching correspondences classification model is  $p_k, k = 1, 2, 3...K$ , so

$$E[F_T(i)] = p_i E[F'_T(i)] \tag{5}$$

$$E[F_T(j)] = (1 - p_j) E[F'_T(j)], i \neq j \tag{6}$$

where  $E$  denotes the mathematical expectation. With the matching correspondences classification integrated, we have

$$p(L_T = i|T) \approx \frac{p_i F'_T(i)W(i)}{p_i F'_T(i)W(i) + \sum_{j=1, j \neq i}^L (1 - p_j) F'_T(j)W(j)} \tag{7}$$

When the precision of each matching correspondences classification model is better than random guess, we have  $p_k > 1/2, k = 1, 2, 3...K$ . It is easy to prove that:

$$p'(\overline{L_T = i|T}) < p(L_T = i|T) \tag{8}$$

At the same time, we have  $\sum_{l=1}^L p'(L_T = l|T) = \sum_{l=1}^L p(L_T = l|T) = 1$ , so it means the matching correspondences classification can enhance the contribution of correct matching correspondences in the confidence inference.

### 5.4 Markov Random Field Integration

Finally, we use a Markov random field to integrate the inferred confidence and spatial smoothness constraint to segment the input image. A graph  $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$  is defined, with each vertex  $p \in \mathcal{V}$  in the graph denotes one super pixel in the input image, while the edges  $\mathcal{E}$  denotes the neighboring relationship between super pixels. Then we build a markov random field upon  $\mathcal{G}$ , with the energy function defined as:

$$E(\mathbf{L}) = \sum_{T \in \mathcal{V}} \psi_i(L_i) + \rho \sum_{e_{ij} \in \mathcal{E}} \psi_{ij}(L_i, L_j) \tag{9}$$

data term  $\psi_i(L_i) = p(L_i|T)$  and smooth term

$$\psi_{ij}(i, j) = [L_i \neq L_j] \frac{1}{1 + \lambda \|D_i - D_j\|^2} \tag{10}$$

where  $D_i$  and  $D_j$  are the feature descriptor of two neighboring super pixels. The alpha expansion algorithm [17] can be used to optimize the energy function and obtain an optimal label configuration.

## 6 Experiments

We used three datasets to test our method: the Google street view dataset used in [10], the CBCL StreetScenes dataset [21] and the Cambridge-driving Labeled Video dataset (CamVid) [22]. They covered street scenes taken from different perspective under different lighting condition (day or dusk). As [8], we will use the category average accuracy (the average proportion of pixels correct in each category) and the global accuracy (total proportion of pixels correct) to evaluate the segmentation performance. For each dataset, 50% images by random selection are put in the database, and the left are used for testing. Our method does not require any 3D geometry information like [10, 11, 23], as it is not limited to the street scenes of image sequences or images of multiple view.

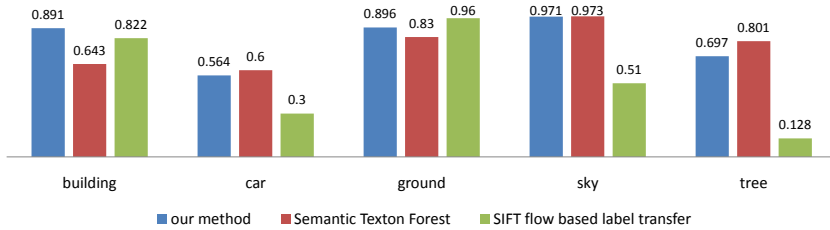
**Parameter setting.** For each dataset, we rescale images so that the average number of pixels contained in each image is about  $320 \times 240$ . For each input image, we select twenty annotated images sets for the KNN-MRF matching. When we decompose an image into super pixels with the method [18], the average size of super pixels is about one hundred pixels. For image level prior prediction and extracting the subset from the database introduced in section 5.1, the top five nearest neighbor by GIST matching are used.

### 6.1 Google Street View Dataset

This dataset consists of about 10,000 images captured in the downtown of Pittsburgh by Google Street View. For evaluation of our method, we randomly select 320 images from this dataset, and labeled them by hand with five categories: *sky*, *ground*, *building*, *car* and *tree*. 160 images are put in the database, and the left 160 images are used for testing. The evaluation includes the following two parts: comparison with two other methods, Semantic Texton Forest [8] and the SIFT flow based label transfer method [13]; and the influence of matching correspondences classification and the influence of different features in the matching correspondences classification models.

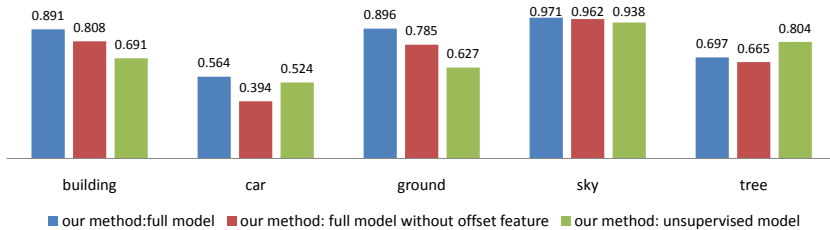
The comparison with [8, 13] is shown in Fig. 3(a). From the comparison result, we found that in terms of category average accuracy and global accuracy, our method is the best among the three methods. On the same dataset, the global accuracy reported in [10] (without 3D geometry and spatial smoothness constraint) is 75.4% (In their evaluation, two more categories: *person* and *recycle bin* with frequency of occurrence under 1% are included).

To analysis how the matching correspondences classification and different features used in matching correspondences classification models influence the performance, we test our method with different setting: full model, full model without offset feature included in matching correspondences classification models and the unsupervised model without matching correspondences classification. The comparison is shown in Fig. 3(b). The comparison result shows that the matching correspondences classification brings significant performance improvement, in terms of category average accuracy and global accuracy. From the comparison of testing with full model and full model without offset feature included



	Our method	Semantic Texton Forest	SIFT flow based label transfer
Global accuracy	0.884	0.744	0.83
Category average accuracy	0.804	0.769	0.544

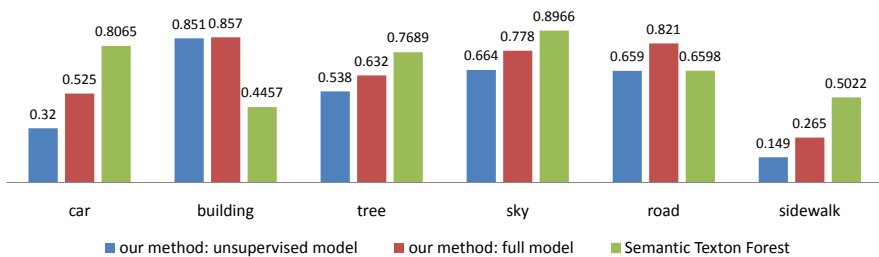
(a)



	full model	full model without offset feature	unsupervised model
Global accuracy	0.884	0.678	0.793
Category average accuracy	0.804	0.717	0.723

(b)

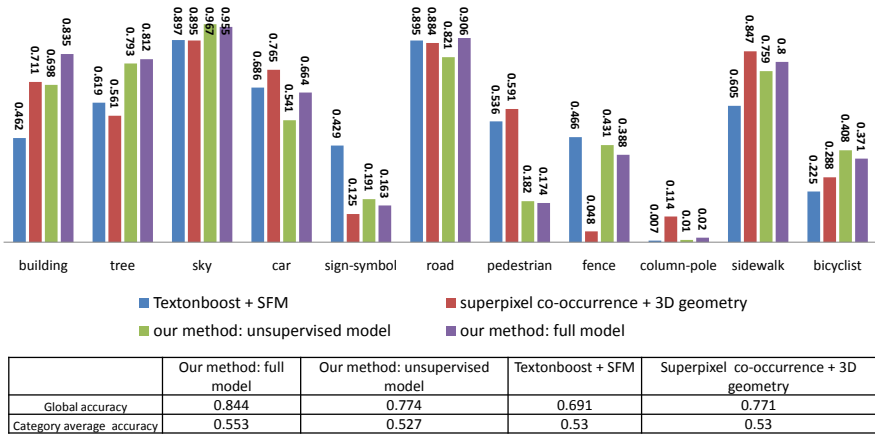
**Fig. 3.** (a) Segmentation accuracy of Our method, Semantic Texton Forest [8] and SIFT flow based label transfer [13] on the Google street view dataset; (b) Segmentation accuracy of our method with different setting on the Google street view dataset



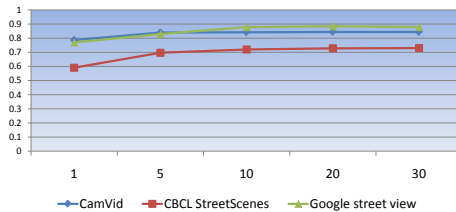
	Our method: full model	Our method: unsupervised model	Semantic Texton Forest
Global accuracy	0.728	0.627	0.619
Category average accuracy	0.646	0.53	0.68

**Fig. 4.** Segmentation accuracy of our method and Semantic Texton Forest [8] on the CBCL StreetScenes dataset

in matching correspondences classification models, we found that the appearance feature and offset feature both contribute to the matching correspondences classification models. Some segmentation examples obtained by our method are given in Fig. 7(a).



**Fig. 5.** Segmentation accuracy of our method, Textonboost + SFM [11] and Superpixel co-occurrence + 3D geometry [23] on the CamVid dataset



**Fig. 6.** Global accuracy(vertical axis) achieved by our method with different number of image sets(horizontal axis) used for confidence inference on different datasets

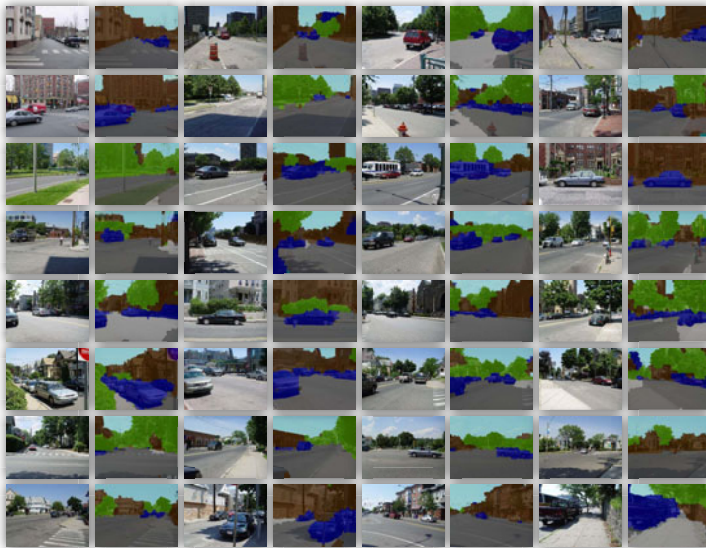
### 6.2 CBCL StreetScenes Dataset

The CBCL StreetScenes dataset contains total 3547 still images of street scenes with annotation, which mainly includes nine categories: *car*, *pedestrian*, *bicycle*, *building*, *tree*, *sky*, *road*, *sidewalk*, and *store*. In our test, the three categories with frequency of occurrence under 1%: *pedestrian*, *bicycle* and *store* are not included. We compared our method with one of the state-of-the-art semantic segmentation techniques: Semantic Texton Forest [8], and the comparison result is given in Fig. 4. The SIFT flow based label transfer method [13] is not included in the comparison, as the time cost of running it on the same train/test split is too high, over 800 hundred hours on a single computer by using the code the authors provided. In terms of category average accuracy, Semantic Texton Forest [8] is better than ours. However, the global accuracy of our method is about 11% higher than that of Semantic Texton Forest [8]. Same as that we found in the test on the previous dataset, the matching correspondences classification improved the performance of our method significantly. Some segmentation examples obtained by our method are given in Fig. 7(b).



Building ■ Car ■ Ground ■ Sky ■ Tree ■

(a) Google street view dataset



Building ■ Car ■ Road ■ Sky ■ Tree ■ Sidewalk ■

(b) CBCL StreetScenes dataset

**Fig. 7.** Some segmentation examples obtained by our method: (a) Google street view dataset;(b) CBCL StreetScenes dataset



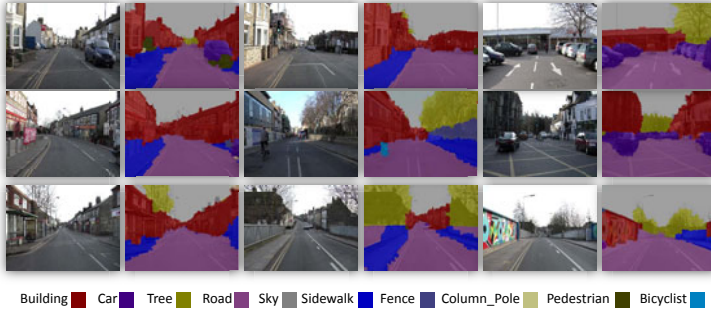


Fig. 8. Some segmentation examples obtained by our method on the CamVid dataset

### 6.3 CamVid Dataset

This dataset provides 701 still images with annotation under different lighting condition, which are extracted from video sequences. To compare with [11, 23] which used the same dataset for testing, we grouped the original label into 11 larger categories and downsampled the images to  $320 \times 240$  pixels as [23]. The result obtained by our method and the result reported in [11, 23] is given in Fig. 5. From the comparison result we found that though no 3D geometry features are used in our method, our method still outperforms [11, 23]. At the same time, we found when no matching correspondences classification models are used, the performance of our method is still close to the state-of-the-art result reported [11, 23]. Last, the same as what we found in the testing on the previous two datasets, integrating matching correspondences classification into the label transfer brings a significant performance improvement. Some segmentation examples obtained by our method are given in Fig. 8.

### 6.4 Computation Time

The time cost of segmenting an input image depends on the size of the corresponding two graphs to be matched with the KNN-MRF matching and how many image sets are used for confidence inference. With our parameter setting, the average time cost for a single KNN-MRF matching between two graphs with average one thousand nodes is about one second. For all the three datasets, the average time cost to segment an input image with our method is under one minute. The global accuracy achieved by our method with different number of image sets used for confidence inference is given in Fig. 6.

## 7 Conclusion

We propose a supervised label transfer method for semantic segmentation of street scenes in this paper. Following the label transfer idea, given an input image, we first find multiple proper image sets from the database, each of which

can cover all semantic categories in the input image. Dense correspondence is established on super-pixel level between the input image and each found image sets with a proposed KNN-MRF matching scheme. Then it is followed by a matching correspondences classification that tries to reduce semantically incorrect correspondence with a supervised learning based method. With those matching correspondences classified as semantically correct correspondence, we infer the confidence value of each super pixel belonging to different semantic categories, and obtain the semantic segmentation of the input image by integrating the inferred confidence value and spatial smoothness constraint in a Markov random field. Experiments show encouraging performances on three standard datasets.

**Acknowledgements.** This work was partially supported by the Hong Kong RGC GRF 618908 and RGC GRF 619409, and the National Natural Science Foundation of China (60933006).

## References

1. Bileschi, S.: StreetScenes: Towards Scene Understanding in Still Images. PhD thesis, Massachusetts Institute of Technology (2006)
2. Xiao, J., Hays, J., Ehinger, K., Oliva, A., Torralba, A.: SUN database: Large scale scene recognition from abbey to zoo. In: IEEE Conference on Computer Vision and Pattern Recognition (2010)
3. Xiao, J., Fang, T., Zhao, P., Lhuillier, M., Quan, L.: Image-based street-side city modeling. *ACM Transactions on Graphics* 28, 114:1–114:12 (2009)
4. Zhao, P., Fang, T., Xiao, J., Zhang, H., Zhao, Q., Quan, L.: Rectilinear parsing of architecture in urban environment. In: IEEE Conference on Computer Vision and Pattern Recognition (2010)
5. Xiao, J., Fang, T., Tan, P., Zhao, P., Ofek, E., Quan, L.: Image-based façade modeling. *ACM Transactions on Graphics* 27, 161:1–161:10 (2008)
6. Tan, P., Fang, T., Xiao, J., Zhao, P., Quan, L.: Single image tree modeling. *ACM Transactions on Graphics* 27, 108:1–108:7 (2008)
7. He, X., Zemel, R., Carreira-Perpinan, M.: Multiscale conditional random fields for image labeling. In: IEEE Conference on Computer Vision and Pattern Recognition (2004)
8. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
9. Shotton, J., Winn, J., Rother, C., Criminisi, A.: TextonBoost for image understanding: Multi-Class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision* 81, 2–23 (2009)
10. Xiao, J., Quan, L.: Multiple view semantic segmentation for street view images. In: IEEE International Conference on Computer Vision (2009)
11. Brostow, G., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and recognition using structure from motion point clouds. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part I. LNCS*, vol. 5302, pp. 44–57. Springer, Heidelberg (2008)

12. Torralba, A., Fergus, R., Freeman, W.: 80 million tiny images: a large dataset for non-parametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(11), 1958–1970 (2008)
13. Liu, C., Yuen, J., Torralba, A.: Nonparametric scene parsing: label transfer via dense scene alignment. In: *IEEE Conference on Computer Vision and Pattern Recognition* (2009)
14. Russell, B., Efros, A., Sivic, J., Freeman, W., Zisserman, A.: Segmenting scenes by matching image composites. In: *Advances in Neural Information Processing Systems* (2009)
15. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision* 42, 145–175 (2001)
16. Bengoetxea, E.: *Inexact Graph Matching Using Estimation of Distribution Algorithms*. PhD thesis, Ecole Nationale Supérieure des Télécommunications (2002)
17. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*. LNCS, vol. 2352, pp. 65–81. Springer, Heidelberg (2002)
18. Levinshtein, A., Stere, A., Kutulakos, K., Fleet, D., Dickinson, S., Siddiqi, K.: Turbopixels: Fast superpixels using geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 2290–2297 (2009)
19. Moosmann, F., Triggs, B., Jurie, F.: Fast discriminative visual codebooks using randomized clustering forests. In: *Advances in Neural Information Processing Systems* (2006)
20. Russell, B., Torralba, A., Liu, C., Fergus, R., Freeman, W.: Object recognition by scene alignment. In: *Object Recognition by Scene Alignment*. *Advances in Neural Information Processing Systems* (2007)
21. Bileschi, S.: CBCL streetscenes challenge framework (2007), <http://cbcl.mit.edu/software-datasets/streetscenes/>
22. Brostow, G., Fauqueur, J., Cipolla, R.: Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters* 30(2), 88–97 (2009)
23. Micusik, B., Kosecka, J.: Semantic segmentation of street scenes by superpixel co-occurrence and 3d geometry. In: *IEEE Workshop on Video-Oriented Object and Event Classification (VOEC)* (2009)

# Category Independent Object Proposals

Ian Endres and Derek Hoiem

Department of Computer Science  
University of Illinois at Urbana-Champaign  
{iendres2,dhoiem}@uiuc.edu

**Abstract.** We propose a category-independent method to produce a bag of regions and rank them, such that top-ranked regions are likely to be good segmentations of different objects. Our key objectives are completeness and diversity: every object should have at least one good proposed region, and a diverse set should be top-ranked. Our approach is to generate a set of segmentations by performing graph cuts based on a seed region and a learned affinity function. Then, the regions are ranked using structured learning based on various cues. Our experiments on BSDS and PASCAL VOC 2008 demonstrate our ability to find most objects within a small bag of proposed regions.

## 1 Introduction

Humans have an amazing ability to localize objects without recognizing them. This ability is crucial because it enables us to quickly and accurately identify objects and to learn more about those we cannot recognize.

In this paper, we propose an approach to give computers this same ability for category-independent localization. Our goal is to automatically generate a small number of regions in an image, such that each object is well-represented by at least one region. If we succeed, object recognition algorithms would be able to focus on plausible regions in training and improve robustness to highly textured background regions. The recognition systems may also benefit from improved spatial support, possibly leading to more suitable coordinate frames than a simple bounding box. Methods are emerging that can provide descriptions for unknown objects [1,2], but they rely on being provided the object's location. The ability to localize unknown objects in an image would be the first step toward having a vision system automatically discover new objects.

Clearly, the problem of category-independent object localization is extremely challenging. Objects are sometimes composed of heterogeneous colors and textures; they vary widely in shape and may be heavily occluded. Yet, we have some cause for hope. Studies of the human visual system suggest that a functioning object localization system can exist in the absence of a functioning object identification system. Humans with damage to temporal cortex frequently exhibit a profound inability to name objects presented to them, and yet perform similar to healthy controls in tasks that require them to spatially manipulate objects [3]. Many objects are roughly homogeneous in appearance, and recent work [4] demonstrates that estimated geometry and edges can often be used to recover occlusion boundaries for free-standing objects. While we cannot expect to localize every object, perhaps we can at least produce a small bag of proposed regions that include most of them.

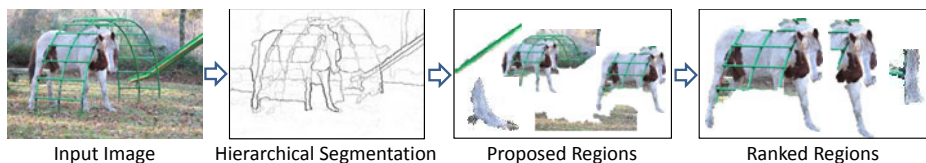
Our strategy is to guide each step of the localization process with estimated boundaries, geometry, color, and texture. First, we create seed regions based on the hierarchical occlusion boundaries segmentation [4]. Then, using these seeds and varying parameters, we generate a diverse set of regions that are guided toward object segmentations by learned affinity functions. Finally, we take a structured learning approach to rank the regions so that the top-ranked regions are likely to correspond to different objects. We train our method on segmented objects from the Berkeley Segmentation Dataset (BSDS) [5], and test it on BSDS and the PASCAL 2008 segmentation dataset [6]. Our experiments demonstrate our system’s ability for category-independent localization in a way that generalizes across datasets. We also evaluate the usefulness of various features for generating proposals and the effectiveness of our structured learning method for ranking.

## 2 Related Work

Here, we relate our work to category-dependent and category-independent methods for proposing object regions.

**Category Dependent Models:** By far, the most common approach to object localization is to evaluate a large number of windows (e.g., [7,8]), which are found by searching naively over position and scale or by voting from learned codewords [9,10], distinctive keypoints [11,12], or regions [13]. These methods tend to work well for objects that can be well-defined according to a bounding box coordinate frame when sufficient examples are present. However, this approach has some important drawbacks. First, it is applicable only to trained categories, so it does not allow the computer to ask “What is this?” Second, each new detector must relearn to exclude a wide variety of textured background patches and, in evaluation, must repeatedly search through them. Third, these methods are less suited to highly deformable objects because efficient search requires a compact parameterization of the object. Finally, the proposed bounding boxes do not provide information about occlusion or which pixels belong to the object. These limitations of the category-based, window-based approach supply some of the motivation for our own work. We aim to find likely object candidates, independent of their category, which can then be used by many category models for recognition. Our proposed segmented regions provide more detail to any subsequent recognition process and are applicable for objects with arbitrary shapes.

**Segmentation and Bags of Regions:** Segmentation has long been proposed as a pre-process to image analysis. Current algorithms to provide a single bottom-up segmentation (e.g., [14,15]) are not yet reliable. For this reason, many have proposed creating hierarchical segmentations (e.g., [16,17]) or multiple overlapping segmentations (e.g., [18,19,20,21]). Even these tend not to reliably produce good object regions, so Malisiewicz et al. [19] propose to merge pairs and triplets of adjacent regions, at the cost of producing hundreds of thousands of regions. In our case, the goal is to segment only objects, such as cars, people, mugs, and animals, which may be easier than producing perceptually coherent or semantically valid partitionings of the entire image. This focus enables a learning approach, in which we guide segmentation and proposal ranking with trained classifiers.



**Fig. 1.** Our pipeline: compute a hierarchical segmentation, generate proposals, and rank proposed regions. At each stage, we train classifiers to focus on likely object regions and encourage diversity among the proposals, enabling the system to localize many types of objects. See section 3 for a more detailed overview.

An alternative approach is to attempt to segment pixels of foreground objects [22] or salient regions [23,24]. However, these approaches may not be suitable for localizing individual objects in cluttered scenes, because a continuous foreground or salient region may contain many objects.

Two concurrent works have also considered generating object proposals as a preprocess for later stages of classification. First, Alexe et al. [25] consider an “objectness” measure over bounding boxes, which they use to bias a sampling procedure for potential object bounding boxes. However, they are limited to the restricted expressiveness of a bounding box. Alternatively, Carreira and Sminchisescu [26] consider a similar region proposal and ranking pipeline to ours. Segmentations are performed using graph cuts and simple color cues, and the regions are ranked through classification based on gestalt cues with a simple diversity model. Our approach guides segmentation with a learned affinity function, rather than setting the image border to background. We also differ in our structured learning approach to diverse ranking.

To summarize our contributions: 1) we incorporate boundary and shape cues, in addition to low-level cues to generate diverse *category independent* object region proposals, and 2) introduce a trained ranking procedure that produces a small diverse set of proposals that aim to cover *all* objects in an image. We thoroughly evaluate each stage of the process, and demonstrate that it can generalize well across datasets for a variety of object categories.

### 3 Overview of Approach

Since our goal is to propose candidates for *any* object in an image, each stage of our process must encourage diversity among the proposals, while minimizing the number of candidates to consider. Our procedure is summarized in Figure 1. To generate proposals for objects of arbitrary shape and size, we adopt a segmentation based proposal mechanism that is encouraged to only propose regions from objects.

Rather than considering only local color, texture, and boundary cues, we include long range interactions between regions of an image. We do this by considering the affinity for pairs of regions to lie on the same object. This set of regions is chosen from a hierarchical segmentation computed over occlusion boundaries. To generate a proposal, we choose one of these regions to seed the segmentation, and compute the probability that each other region belongs to the same object as this seed. The affinities are then transferred to a graph over superpixels from which we compute segmentations with a variety of parameters. By computing

the affinities over regions first and then transferring them to superpixels, we get the benefit of more reliable predictions from larger regions while maintaining the flexibility of a superpixel based segmentation. After repeating this process for all seed regions, we obtain an initial bag of proposals.

In our effort to discover a diverse set of objects, our proposal mechanism may generate many redundant or unlikely object candidates. In both cases, we would like to suppress these undesirable proposals, allowing us to consider better candidates first. This motivates a ranking procedure that provides an ordering for a bag of proposals which simultaneously suppresses both redundant and unlikely candidates. We can then uncover a diverse set of the good object proposals with far fewer candidates.

Our ranker incrementally adds proposals, from best to worst, based on the combination of an object appearance score and a penalty for overlapping with previously added proposals. By taking into account the overlap with higher ranked proposals, our ranker ensures that redundant regions are suppressed, forcing the top ranked regions to be diverse. This is especially important in images with one dominant object and several “auxiliary” objects.

## 4 Proposing Regions

We first generate a large and diverse bag of proposals that are directed to be more likely to be object regions. Each proposal is generated from a binary segmentation, which is seeded with a subregion of the image. This seed is assumed to be foreground, and a segmenter selects pixels likely to belong to the same foreground object as the seed.

### 4.1 Hierarchical Segmentation

We use regions and superpixels from a hierarchical segmentation as the building blocks for our proposal mechanism. To generate the hierarchical segmentation, we use the output of the occlusion boundary algorithm from Hoiem et al. [4] (the details of this algorithm are not relevant to our paper). The occlusion algorithm outputs four successively coarser segmentations, with a probability of occlusion and of the figure/ground label for each boundary in the segmentation. From each segmentation, we compute a probability of boundary pixel map and a figure/ground probability pixel map, and we average over the segmentations. Then, we create our hierarchical segmentation with agglomerative grouping based on boundary strength, as in [16], and we use the boundary strength and figure/ground likelihoods as features.

### 4.2 Seeding

A seed serves as the starting point for an object proposal. The appearance and boundaries around the seed are used to identify other regions that might belong to the same object. Seeds are chosen from the hierarchical segmentation such that they are large enough to compute reliable color and texture distributions. Also, we remove regions with boundaries weaker than 0.01, since these are likely to just be a portion of a larger region. Stronger boundaries also facilitate the use of boundary cues to determine the layout of the object with respect to the regions.

### 4.3 Generating Segmentations

**CRF Segmentation:** To generate a proposal, we infer a foreground / background labeling  $\mathbf{l}, l_i \in \{0, 1\}$  over superpixels. Given a seed region, defined by a set of superpixels  $S$ , we construct a CRF that takes into account each superpixel’s affinity for the seed region and the probability of boundaries between adjacent superpixels:

$$P(\mathbf{l}|X, S, \gamma, \beta) \propto \exp \left( \sum_i f(l_i; S, X, \gamma) + \beta \sum_{\{i,j\} \in N} g(l_i, l_j; X) \right) \quad (1)$$

Here,  $f(l_i; S, X, \gamma)$  is the superpixel affinity term, inferred from image features  $X$ , and  $g(l_i, l_j; X)$  is the edge cost between adjacent superpixels (defined by set of neighbors  $N$ ). This CRF is parametrized by the foreground bias  $\gamma$  and the affinity/edge trade-off  $\beta$ . By varying these parameters for each seed, we can produce a more diverse set of proposals. We choose five  $\gamma$  values from between  $[-2, 2]$ , and five  $\beta$  values from  $[0, 5]$ .

**Affinity:** To compute the superpixel affinity  $f(l_i; S, X, \gamma)$ , we first compute each region  $R$ ’s affinity for lying on the same object as the seed  $S$ . We learn the foreground probability  $P(l_R|S, X)$  with a boosted decision tree classifier. Positive training examples are generated from pairs of regions that lie on the same object. Negative examples use pairs with one region lying on an object, and the other region lying on another object or the background.

The classifier uses features for cohesion, boundary, and layout cues, as summarized in Table 1. *Cohesion* is encoded by the histogram intersection distances of color and texture ( $P1$ ). *Boundary cues* are encoded by considering the cost to pass across boundaries from one region to the other. This path across boundaries is the straight line between their centers of mass ( $P2$ ).

**Table 1.** Features computed for pairs of regions for predicting the likelihood that the pair belongs to the same object. These features can capture non-local interactions between regions, producing better segmentations.

<i>Feature Description</i>	<i>Length</i>
P1. Color,Texture histogram intersection	2
P2. Sum,Max strength of boundary crossed between centers of mass	2
L1. Left+Right layout agreement	1
L2. Top+Bottom layout agreement	1
L3. Left+Right+Top+Bottom layout agreement	1

We introduce a new layout feature. Given occlusion boundaries and figure/ground labels, we predict whether a particular region is on the left, right, top, bottom, or center of the object. These predictions are made by boosted decision tree classifiers based on histograms of occlusion boundaries, where the boundaries are separated based on figure/ground labels. As a feature, we measure whether the layout predictions for two regions are consistent with them being on the same object. For example, if one region predicts that it is on the left of the object and a second region to the right of the first predicts that it



is on the right side of the object, those regions are consistent. We construct a *layout* score for horizontal, vertical, and overall agreement ( $L1 - L3$ ).

Since the CRF is defined over superpixels, the region affinity probabilities are transferred to each superpixel  $i$  by averaging over the regions that contain it. The terms of this average are weighted by the probability that each region  $R$  is homogeneous ( $P(H_R)$ ), which is predicted from the appearance features in Table 2:

$$P(l_i = 1|S, X) = \frac{\sum_{\{R|i \in R\}} P(H_R) \cdot P(l_R = 1|S, X)}{\sum_{\{R|i \in R\}} P(H_R)}. \tag{2}$$

Note that we now have labels for superpixels ( $l_i$ ) and for regions ( $l_R$ ). We use  $P(l_i|S, X)$  to compute the affinity term  $f(l_i; S, X, \gamma)$ :

$$f(l_i; S, X, \gamma) = \begin{cases} 0 & : l_i = 1, i \in S \\ \infty & : l_i = 0, i \in S \\ -\ln \left( \frac{P(l_i=0|X)}{P(l_i=1|X)} \right) + \gamma & : l_i = 1, i \notin S \end{cases} \tag{3}$$

The infinite cost ensures that superpixels belonging to the seed are labeled foreground.

**Edge Cost:** The edge cost enforces a penalty for assigning different labels to adjacent superpixels when their separating boundary is weak. This boundary strength is computed from the occlusion boundary estimates for each pair of adjacent superpixels  $i, j$ :  $P(B_{i,j}|X)$ .

$$g(l_i, l_j; X) = \begin{cases} 0 & : l_i = l_j \\ -\ln P(B_{i,j}|X) & : l_i \neq l_j \end{cases} \tag{4}$$

This edge cost produces a submodular CRF, so exact inference can be computed quickly with a single graph-cut [27] for each seed and parameter combination. Proposals with disconnected components are split, and highly overlapping ( $\geq 97\%$ ) proposals are pruned. Further non-maximum suppression is handled in the ranking stage.

## 5 Ranking Proposals

We now introduce a ranker that attempts to order proposals, such that each object has a highly ranked proposal. This ranker encourages diversity in the proposals allowing us to achieve our goal of discovering *all* of the objects in the image. Below, we detail our objective function, which encourages top-ranked regions to correspond to different objects and more accurate object segmentations to be ranked higher. Then, we explain the image features that we use to rank the regions. Finally, we describe the structured learning method for training the ranker.

**Formulation:** By writing a scoring function  $S(\mathbf{x}, \mathbf{r}; \mathbf{w})$  over the set of proposals  $\mathbf{x}$  and their ranking  $\mathbf{r}$ , we can take advantage of structured learning. The goal is to find the parameters  $\mathbf{w}$  such that  $S(\mathbf{x}, \mathbf{r}; \mathbf{w})$  gives higher scores to rankings that place proposals for all objects in high ranks.

$$S(\mathbf{x}, \mathbf{r}; \mathbf{w}) = \sum_i \alpha(r_i) \cdot \left( \mathbf{w}_a^T \Psi(x_i) - \mathbf{w}_p^T \Phi(r_i) \right) \tag{5}$$

The score is a combination of appearance features  $\Psi(x)$  and overlap penalty terms  $\Phi(r)$ , where  $r$  indicates the rank of a proposal, ranging from 1 to the number of proposals  $M$ . This allows us to jointly learn the appearance model and the trade-off for overlapping regions.  $\Phi_1(r)$  penalizes regions with high overlap with previously ranked proposals, and  $\Phi_2(r)$  further suppresses proposals that overlap with *multiple* higher ranked regions. The second penalty is necessary to continue to enforce diversity after many proposals have at least one overlapping proposal:

$$\Phi_1(r_i) = \max_{\{j|r_j < r_i\}} ov(i, j) \quad (6)$$

$$\Phi_2(r_i) = \sum_{\{j|r_j < r_i\}} ov(i, j) \quad (7)$$

The overlap score is computed as the area of two regions' intersection divided by their union, with  $A_i$  indicating the set of pixels belonging to region  $i$ :

$$ov(i, j) = \frac{|A_i \cap A_j|}{|A_i \cup A_j|} \quad (8)$$

Each proposal's score is weighted by  $\alpha(r)$ , a monotonically decreasing function. Because higher ranked proposals are given more weight, they are encouraged to have higher scores. We found that the specific choice of  $\alpha(r)$  is not particularly important, as long as it falls to zero for a moderate rank value. We use  $\alpha(r) = \exp\left(\frac{(r-1)^2}{\sigma^2}\right)$ , with  $\sigma = 150$ .

Computing  $\max_{\mathbf{r}} S(\mathbf{x}, \mathbf{r}; \mathbf{w})$  cannot be solved exactly, so we use a greedy approximation that incrementally adds the proposal with the maximum marginal gain. We found that this works well for a test problem where full enumeration is feasible, especially when  $ov(\cdot, \cdot)$  is sparse, which is true for this ranking problem.

**Representation:** The appearance features  $\Psi(x)$  characterize general properties for typical object regions, as summarized in Table 2. Since this is a category independent ranker, we cannot rely on finely tuned category dependent shape and appearance models. However, we can expect object boundaries to respect occlusion boundaries, so we encode the probability that the exterior is occluded by or occluding another region. We also encode the probability of interior boundaries, which we expect to be small.

Additionally, certain "stuff-like" regions can be quickly identified as background, such as grass and sidewalks, so we learn a pixel based probability of background classifier on LabelMe [28], and characterize the response within the region. We also use the confidence of the vertical solid non-planar geometric class, using trained classifiers from [29], which is noted to often correspond to object classes. Finally, we encode the differences between color and texture distributions between the object and background. We compute the difference in histograms between the object and two regions: the local background region surrounding the object and the entire background.

**Learning:** To solve the structured learning problem, we use the slack-rescaled method with loss penalty used in [30]. This method finds the highest scoring

**Table 2.** Features used to describe the appearance of a proposal region. It is important that each of these features generalize across all object categories, including ones never seen during training.

<i>Feature Description</i>	<i>Length</i>
B1. Mean,max probability of exterior boundary	2
B2. Mean,max probability of interior boundary	2
B3. Mean,max probability that exterior occludes	2
B4. Mean,max probability of exterior being occluded	2
S1. Min,mean,max,max-min probability of background	4
S2. Min,mean,max,max-min probability of vertical surface	4
S3. Color,texture histogram intersection with local background	2
S4. Color,texture histogram intersection with global background	2

labeling, rather than the most violated constraint, and adds an additional cost to the objective to penalize for high loss candidates:

$$\begin{aligned}
 \min_{\mathbf{w}, \xi_n} \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C_1}{N} \sum_n \xi_n + \frac{C_2}{N} \sum_n \mathcal{L}(\mathbf{r}^{(n)}, \hat{\mathbf{r}}^{(n)}) \quad (9) \\
 \text{s.t. } \forall \mathbf{r} \in P^{(n)} \setminus \mathbf{r}^{(n)}, \forall n \\
 S(\mathbf{x}^{(n)}, \mathbf{r}^{(n)}; \mathbf{w}) - S(\mathbf{x}^{(n)}, \mathbf{r}; \mathbf{w}) \geq 1 - \frac{\xi_n}{\mathcal{L}(\mathbf{r}^{(n)}, \mathbf{r})}, \\
 \xi_n \geq 0 \\
 \mathbf{w}_p \geq 0
 \end{aligned}$$

where, for image  $n$ ,  $\mathbf{r}^{(n)}$  is the ground truth ranking,  $\hat{\mathbf{r}}^{(n)} = \operatorname{argmax}_{\mathbf{r} \in P^{(n)}} S(\mathbf{x}^{(n)}, \mathbf{r}; \mathbf{w})$  is the highest scoring proposal, and  $P^{(n)}$  is the set of valid labellings, in this case, the set of permutations over regions. The cutting plane approach avoids having to exhaustively enumerate the resulting intractable set of constraints.

The loss  $\mathcal{L}$  must enforce two properties: higher quality proposals should have higher ranks ( $\mathcal{L}_1$ ), and each object  $o$  in the set of objects  $O$  should have a highly ranked proposal ( $\mathcal{L}_2$ ):

$$\begin{aligned}
 \mathcal{L}(\mathbf{r}, \hat{\mathbf{r}}) &= \frac{1}{2} \mathcal{L}_1(\mathbf{r}, \hat{\mathbf{r}}) + \frac{1}{2} \mathcal{L}_2(\mathbf{r}, \hat{\mathbf{r}}) \\
 \mathcal{L}_1(\mathbf{r}, \hat{\mathbf{r}}) &= \frac{1}{|O|} \sum_{o \in O} \sum_{\{(i,j) | r_i < r_j\}} I[ov(i, o) < ov(j, o)] \\
 \mathcal{L}_2(\mathbf{r}, \hat{\mathbf{r}}) &= \frac{1}{|O|} \sum_{o \in O} \min_{\{i | ov(i, o) \geq 50\% \}} r_i
 \end{aligned} \quad (10)$$

To learn this structured model, we iteratively find the highest scoring ranking for an image, update  $\mathbf{w}$  with this new constraint, and repeat until the change in  $\mathbf{w}$  is small.

## 6 Experiments and Results

We perform experiments on the Berkeley Segmentation Dataset (BSDS) [5] and the Segmentation Taster images from PASCAL VOC 2008 [6]. All training and parameter selection is performed on the BSDS training set, and results are evaluated on BSDS test and the PASCAL validation set. For both datasets, a ground truth segmentation is provided for each object. For BSDS, we label object regions by merging the original ground truth segments so that they correspond to objects.

Qualitative results from both PASCAL and BSDS are sampled in Figure 2.



**Fig. 2.** Results from the proposal and ranking stages on BSDS (first 3 rows) and PASCAL 2008 (last 3 rows). The left column shows the 3 highest ranked proposals, The center column shows the highest ranked proposal with 50% overlap for each object. The right column shows the same for a 75% threshold. The number pairs displayed on each proposal correspond to rank and overlap, respectively. The desk scene demonstrates the diversity of our ranking. The train and deer demonstrate the high quality of proposals.

## 6.1 Proposal Generation

To measure the quality of a bag of proposals, we find the best segmentation overlap score for each object (BSS). From this, we can characterize the overall quality of segments with the mean BSS over objects, or compute the recall by thresholding the BSS at some value, and counting the number of objects with a BSS of at least this threshold. For our experiments, we set the threshold to 50% unless otherwise noted. A pixel-wise overlap threshold of 50% is usually, but not always, more stringent than a 50% bounding box overlap.

**Features:** The most commonly used features for segmentation are color and texture similarity, so we use this as a baseline. We then add the boundary crossing and layout features individually to see their impact. Finally, we combine all of the features to obtain our final model. To measure the performance of each feature, we consider the area under the ROC curve (AUC) for affinity classification, the best segment score, and recall at 50%. The results are shown in Table 3.

The first thing to note is that the addition of both the boundary and layout features are helpful for both datasets. In addition, we find that the affinity classification performance cannot fully predict a feature’s impact on proposal performance. It is important to also consider how well the features facilitate producing a diverse set of proposals. Features that cause prediction to be more dependent on the seed region will produce a more diverse set of proposals.

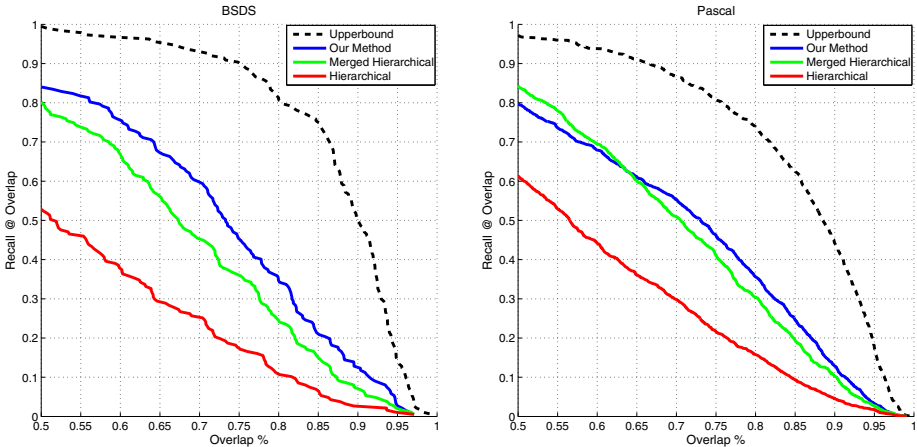
**Table 3.** A comparison of how features impact affinity classification (AUC), recall @ 50% overlap, and best segment score (BSS). Both classification accuracy and diversity of proposals must be considered when choosing a set of features.

Feature	BSDS			PASCAL		
	AUC	Recall	BSS	AUC	Recall	BSS
Color,Texture (P1)	0.72	75.4 %	0.655	0.68	78.8%	0.67
C,T + Boundary Crossing (P1,P2)	0.77	81.8%	0.671	0.76	79.7%	<b>0.68</b>
C,T + Layout (P1,L1,L2,L3)	0.74	82.9%	0.679	0.71	<b>81.1%</b>	<b>0.68</b>
All (P1,P2,L1,L2,L3)	<b>0.83</b>	<b>84.0%</b>	<b>0.69</b>	<b>0.80</b>	79.7%	<b>0.68</b>

**Proposal Quality:** We define similar baselines to [19]. The first baseline is to use each region from the hierarchical segmentation as an object proposal. The second baseline is to merge all pairs of adjacent regions, which achieves higher recall but with many more proposals. We can also measure the upper bound on performance by choosing the best set of superpixels for each object region.

It is clear from Figure 3 that the initial hierarchical segmentation is not well suited for proposing object candidates. After merging proposals, the segmentation quality is comparable to our method, but as Figure 6 shows, it produces more than an order of magnitude more proposals. For both datasets, our method produces more high quality proposals for overlaps greater than 65%.

Finally, we provide a breakdown of recall for individual categories of the PASCAL dataset in Figure 4. These results are especially promising, because many of the categories with high recall, such as dog and cat, are difficult for standard detectors to locate. The low performance for categories like car and sheep is mainly due to the difficulty of proposing small regions (< 0.5% of the image area, or < 1000 pixel area), especially when the objects are in crowded scenes. The dependence of recall on area is shown in Figure 5.



**Fig. 3.** These curves characterize the quality of proposals from each method, showing the percentage of objects recalled for a given overlap %. For BSDS, we generate better proposals for all levels of overlap. For PASCAL, we outperform the baselines for higher recall levels and are still comparable at 50% overlap. These results are impressive because we consider 20-30 times fewer regions.

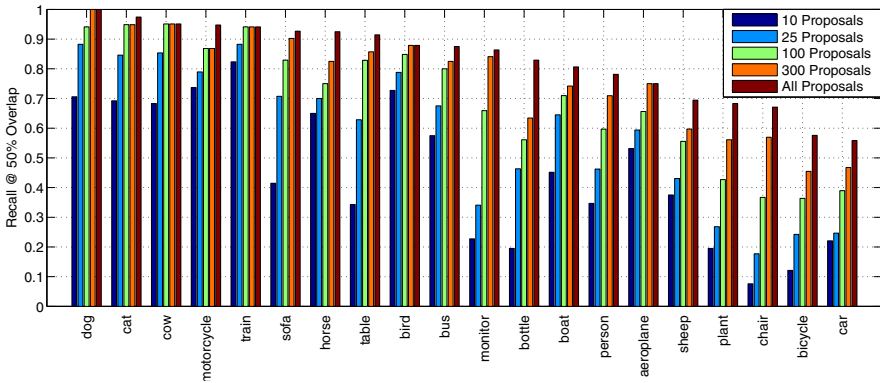
## 6.2 Ranking Performance

We compare our ranking method to three baselines. The first method scores each proposal independently, and the ranking is produced by sorting these scores from high to low, as in [26]. Positive examples are chosen from a pool proposals with at least 50% overlap with some object and negative examples have no more than 35% overlap with any object. The second baseline includes the overlap penalty of our method, but learns the appearance model and trade-off terms separately. The final baseline simply assigns random ranks to each proposal. This can be seen as encouraging diversity without taking into account appearance. To evaluate the quality of our ranker, we measure the number of objects recalled when we threshold each image’s bag at a certain size. The results are presented in Figure 6.

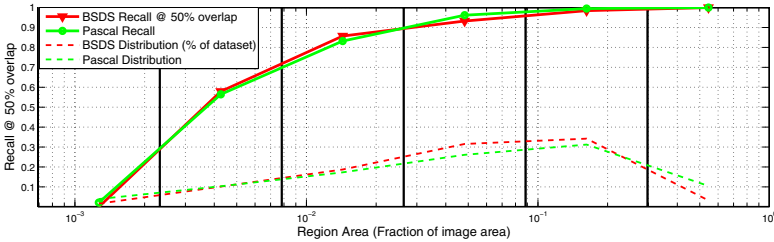
We find that by jointly learning the appearance and suppression models, our method outperforms each of the baselines. Because the independent classifier does not encourage diversity, only the first object or object-like region is given a high rank, and the number of proposals required to recall the remaining objects can be quite high. In fact, when considering more than 10 proposals, the random ranker quickly outperforms the independent classifier. This emphasizes the importance of encouraging diversity. However, both models that include both appearance models and overlap terms outperform the random ranker. Finally, by learning with an appropriate loss and jointly learning the model, we achieve small but noticeable gains over the baseline with an overlap term.

## 7 Discussion

We have introduced a procedure that generates a small, but diverse set of category-independent object proposals. By incorporating the affinity predictions,



**Fig. 4.** Recall for each object category in PASCAL. These results are quite promising because many of the categories with high recall are difficult for standard object detectors to recognize. For many categories, most of the instances can be discovered in the first 100 proposals.

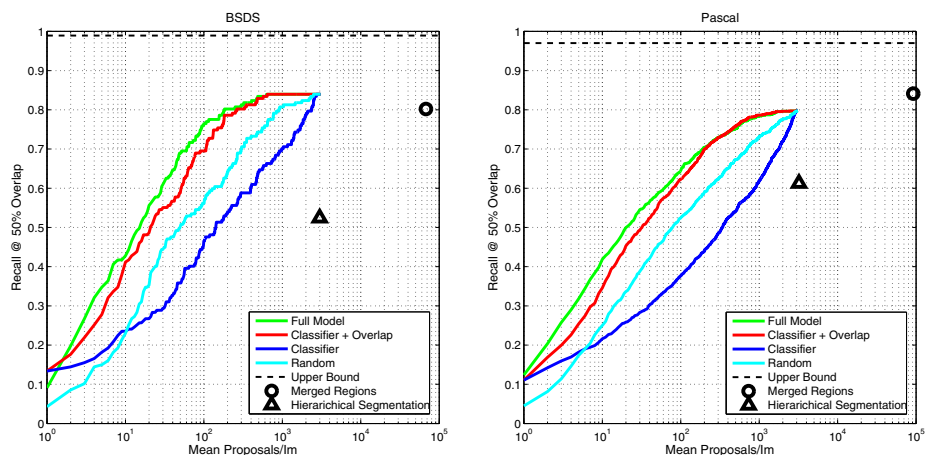


**Fig. 5.** Recall vs. object size: The plot shows the percentage of recalled objects based on their area, relative to the image size. Histogram bin edges are indicated by solid vertical lines. This demonstrates that uncovering smaller objects is more difficult than larger objects, but nearly 60% of objects between 0.3% and 0.8% of the image are still recovered. This is due to weaker object cues and because the region overlap criteria is more sensitive to individual pixel errors for smaller objects. The dashed lines also show the proportions of the dataset for each object size.

we can direct the search for segmentations to produce good candidate regions with far fewer proposals than standard segmentations. Our ranking can further reduce the number of proposals, while still maintaining high diversity. Our experiments show that this procedure generalizes well and can be applied for many categories.

The results on PASCAL are especially encouraging, because with as few as 100 proposals per image, we can obtain high recall for many categories that standard scanning window detectors find difficult. This is quite amazing, considering that the system had never seen most of the PASCAL categories during training!

Beyond categorization, our proposal mechanism can be incorporated in applications where category models are not available. When presented with images of new objects, our proposals can be used in an active learning framework to learn about unfamiliar objects. Alternatively, they can be used for automatic object discovery methods such as [20]. Combined with the description based recognition methods [11,2], we could locate and describe new objects.



**Fig. 6.** Recall vs. number of proposals per image: When considering recall for more than 10 proposals per image, enforcing diversity (Random) is a more important than object appearance (Classifier). Combining diversity and appearance (Classifier + Overlap) improves performance further, and jointly learning both (Full model) gives further gains.

While this method performs well in general, it has difficulty in cases where the occlusion boundary predictions fail and for small objects. These are cases where having some domain knowledge, such as appearance or shape models can complement a generic proposal mechanism. This suggests a joint approach in which bottom-up region proposals are complemented by part or category detectors that incorporate domain knowledge.

## Acknowledgments

This work was supported in part by the National Science Foundation under IIS-0904209.

## References

1. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: CVPR (2009)
2. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: CVPR (2009)
3. Goodale, M.A., Milner, A.D., Jakobson, L.S., Carey, D.P.: A neurological dissociation between perceiving objects and grasping them. *Nature* 349, 154–156 (2000)
4. Hoiem, D., Stein, A.N., Efros, A.A., Hebert, M.: Recovering occlusion boundaries from an image. In: ICCV (2007)
5. Martin, D., Fowlkes, C., Malik, J.: Learning to find brightness and texture boundaries in natural images. In: NIPS (2002)
6. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge, VOC 2008 Results (2008), <http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html>



7. Viola, P., Jones, M.J.: Robust real-time face detection. *IJCV* 57 (2004)
8. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multi-scale, deformable part model. In: *CVPR* (2008)
9. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. *IJCV* 77, 259–289 (2008)
10. Maji, S., Malik, J.: Object detection using a max-margin hough transform. In: *CVPR*, pp. 1038–1045. IEEE Computer Society, Los Alamitos (2009)
11. Chum, O., Zisserman, A.: An exemplar model for learning object classes. In: *CVPR* (2007)
12. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: *ICCV* (2009)
13. Gu, C., Lim, J., Arbelaez, P., Malik, J.: Recognition using regions. In: *CVPR*, pp. 1030–1037 (2009)
14. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Trans. PAMI* 22 (2000)
15. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. *IJCV* 59 (2004)
16. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: From contours to regions: An empirical evaluation. In: *CVPR*, pp. 2294–2301 (2009)
17. Sharon, E., Galun, M., Sharon, D., Basri, R., Brandt, A.: Hierarchy and adaptivity in segmenting visual cues. *Nature* (2006)
18. Hoiem, D., Efros, A.A., Hebert, M.: Geometric context from a single image. In *ICCV* (2005)
19. Malisiewicz, T., Efros, A.A.: Improving spatial support for objects via multiple segmentations. In: *BMVC* (2007)
20. 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: *CVPR* (2006)
21. Stein, A., Stepleton, T., Hebert, M.: Towards unsupervised whole-object segmentation: Combining automated matting with boundary detection. In: *CVPR* (2008)
22. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: *ICCV* (2009)
23. Walther, D., Koch, C.: 2006 special issue: Modeling attention to salient proto-objects. *Neural Networks* 19, 1395–1407 (2006)
24. Liu, T., Sun, J., Zheng, N., Tang, X., Shum, H.: Learning to detect a salient object. In: *CVPR*, pp. 1–8 (2007)
25. Alexe, B., Deselaers, T., Ferrari, V.: What is an object? In: *CVPR* (2010)
26. Carreira, J., Sminchisescu, C.: Constrained parametric min cuts for automatic object segmentation. In: *CVPR* (2010)
27. Rother, C., Kolmogorov, V., Blake, A.: “grabcut”: interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.* 23, 309–314 (2004)
28. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: LabelMe: a database and web-based tool for image annotation. Technical report, MIT (2005)
29. Hoiem, D., Efros, A.A., Hebert, M.: Recovering surface layout from an image. *IJCV* 75, 151–172 (2007)
30. Szummer, M., Kohli, P., Hoiem, D.: Learning crfs using graph cuts. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part II. LNCS*, vol. 5303, pp. 582–595. Springer, Heidelberg (2008)

# Photo-Consistent Planar Patches from Unstructured Cloud of Points

Roberto Toldo and Andrea Fusiello

Dipartimento di Informatica  
Università di Verona  
Strada Le Grazie 15, 37134 Verona, Italy  
{roberto.toldo, andrea.fusiello}@univr.it

**Abstract.** Planar patches are a very compact and stable intermediate representation of 3D scenes, as they are a good starting point for a complete automatic reconstruction of surfaces. This paper presents a novel method for extracting planar patches from an unstructured cloud of points that is produced by a typical structure and motion pipeline. The method integrates several constraints inside J-linkage, a robust algorithm for multiple models fitting. It makes use of information coming both from the 3D structure and the images. Several results show the effectiveness of the proposed approach.

## 1 Introduction

While the current state of the art in architectural three-dimensional (3D) reconstruction has focused on the recovery of dense and accurate representations of objects imaged through pictures or video, the sustained interest in accessible architectural modeling software is a strong evidence of an untapped general need for compact, abstract representations of architectural objects. What separates unstructured cloud of points from higher-level renditions of an architectural model is a *semantic gap*, which should be bridged exploiting additional information. This is one of the most challenging research area in Computer Vision. The proposed methods can be divided in three main categories: interactive, top-down and bottom-up.

Interactive approaches require user intervention to recognize higher level structures, usually basing on the three-dimensional information previously extracted [1–4]. Top-down or model based approaches start from the prior knowledge of the set of potential parametric models and try to infer the best fitting one along with its parameters [5–9]. Potentially, only one image could be employed if the prior knowledge is enough to derive the 3D model [10, 11]. When no prior knowledge is assumed or user intervention is not available, bottom-up methods are employed. They start directly from raw three-dimensional data points trying to aggregate them in progressively higher level structures, possibly using also the information coming from the images. This paper falls in this category: The aim is to leverage models from unorganized point clouds to an intermediate representation,

i.e. planar patches, that narrows the gap between acquisition and manipulation of architectural models.

Some methods try to optimize an initial triangulation using visibility [12] or photo-consistency [13, 14] only, i.e., the fact that a patch corresponding to a solid opaque surface has the same appearance in all the images (modulo some geometric and photometric distortions). They work only with very simple convex polyhedra objects, and they assume all points visible by at least one view. A similar approach was proposed in [15]. A sequential MSAC [16] is employed in order to detect the planes. Image-consistent triangulation is then used within a simulated annealing algorithm to create an optimal surface mesh. In [17] an automatic approach to segment a cloud of points into planes is proposed: It generates plane hypotheses by random sampling the 3D points (inspired by RANSAC) and scores them using photo-consistency. The reported experiments involve extremely simple objects. More recently Moser et al. [18] presented a paper that is able to perform out-of-core simplification of an high quality digital surface model of a city using RANSAC. The density and good quality of the input data are crucial here.

Very recent works proposed to run a Multiview Stereo on the output a SaM pipeline [22, 23]. While the results are visually compelling, they do not point at the problem of the semantic gap, since the output is a dense and less compact representation of the scene.

Besides [12–14] which are very simple, all the above papers share a common part since they extract the planes underlying the scenes using RANSAC (or MSAC) with spatial or photo-consistency information. This seems to be a crucial task, but the sequential application of an algorithm designed for single model extraction, is not suitable, and this becomes clear as soon as one steps from clean, structured data to real, noisy unstructured data, as those coming from a structure and motion (SaM) pipeline [19, 20]. Techniques designed to extract *multiple* instances of a model are required in this case, e.g. J-linkage, which has recently been proposed [21] and proved to be very robust. It will be described in details in section 2.

Our strategy reaps the benefits of most of the aforementioned methods: i) it applies to unorganized large cloud of points, ii) employ a multiple model fitting algorithm (J-linkage) and iii) seamlessly integrates both spatial, visibility *and* photo-consistency information inside it.

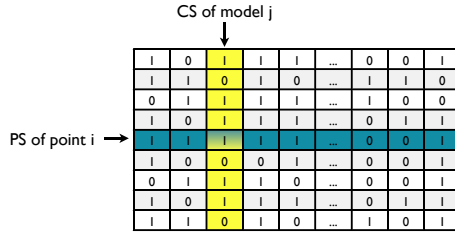
The output of our algorithm are triangulated planar patches, which are a very compact and stable intermediate representation of 3D scenes, as they are a good starting point for a complete automatic reconstruction of surfaces.

## 2 Overview of the J-linkage Algorithm

In this section the J-linkage algorithm will be briefly overviewed. More details can be found in [21].

The method is based on random sampling, like RANSAC. Each minimal sample set (MSS) defines a tentative model. Imagine to build a  $N \times M$  matrix (Fig. 2)

where entry  $(i, j)$  is 1 if point  $i$  is closer to model  $j$  than a threshold  $\varepsilon$ . Each column of that matrix is the characteristic function of the *consensus set* of a model. Each row is the characteristic function of the *preference set* (PS) of a given point, i.e., indicates which models a points has given consensus to. Points belonging to the same structure will have similar PS, in other words, they will cluster in the conceptual space  $\{0, 1\}^M$ .



**Fig. 1.** An example of consensus/preference matrix. Columns are consensus sets (CS), rows are preference sets (PS).

### 2.1 Random Sampling

As in [24] it is assumed that the a-priori probability that two points belong to the same structure is higher the smaller the distance between the points. Hence minimal sample sets are constructed in a way that neighboring points are selected with higher probability. If a point  $\mathbf{x}_i$  has already been selected, then  $\mathbf{x}_j$  has the following probability of being drawn:

$$P(\mathbf{x}_j | \mathbf{x}_i) = \begin{cases} \frac{1}{Z} \exp -\frac{\|\mathbf{x}_j - \mathbf{x}_i\|^2}{\sigma^2} & \text{if } \mathbf{x}_j \neq \mathbf{x}_i \\ 0 & \text{if } \mathbf{x}_j = \mathbf{x}_i \end{cases} \quad (1)$$

where  $Z$  is a suitable normalization constant and  $\sigma$  is chosen heuristically.

### 2.2 Agglomerative Clustering

Models are extracted by agglomerative clustering data points in the conceptual space, where each point is represented by its PS. The distance between two elements (point or cluster) is computed as the *Jaccard* distance between the respective preference sets. The PS of a cluster is defined as the intersection of the preference sets of its points. Given two sets  $A$  and  $B$ , the Jaccard distance is

$$d_J(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}. \quad (2)$$

The Jaccard distance measures the degree of overlap of  $A$  and  $B$  and ranges from 0 ( $A = B$ ) to 1 ( $A \cap B = \emptyset$ ).

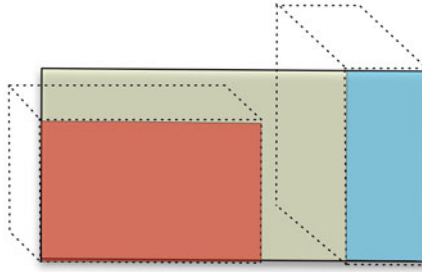
The algorithm proceeds by linking elements with distance smaller than 1 and stops as soon as there are no such elements left. This can be performed efficiently using an heap data structure. As a result, clusters have the following properties:

- for each cluster there exists at least one model that is in the PS of all its points;
- one model cannot be in the PS of *all* the points of two distinct clusters;

The final model parameters for each cluster of points is estimated by least squares fitting.

### 3 Constraints Integration

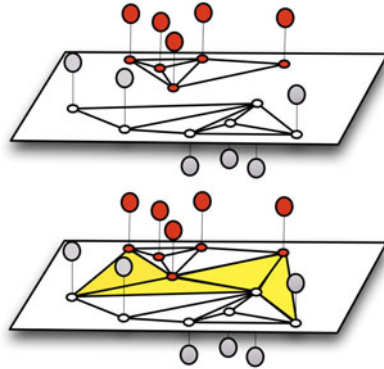
This paper is aimed at leveraging the J-linkage algorithm to fit planar patches to a cloud of 3D point that are samples of surfaces in the observed scene. Extraction of planar patches is not the same as fitting planes, because a patch is a *region* of the plane, and the same plane may contain more patches (see Fig. 2). The planar patch associated to a set of coplanar points is the convex hull of the projection of the points onto the fitting plane. In order for a planar patch to represent an actual surface, it must satisfy a number of constraints, beside coplanarity, that will be described later. This section will concentrate on how these constraints can be seamlessly integrated inside J-linkage.



**Fig. 2.** A single plane (yellow) contains several patches (blue and red)

J-linkage extracts models in an incremental way, by merging smaller structures at each step. In the case on planar patches, two patches can merge only if the result is a set of coplanar points (to some extent). Coplanarity is the invariant property, and any other constraint can be enforced as an invariant property, so that two patches can be merged if and only if the resulting does not violate the constraint.

More in detail, the constraints will be formulated and tested on triangles, since any planar polygon can be triangulated. When two patches are being considered for possible merging, a new patch is computed as the convex hull of the union of the points. By inductive hypothesis the two original patches satisfy the constraints, whereas the new triangles that are created must be tested against the constraints. If a single triangle fails the merging is rejected. A graphical explanation of this incremental step is shown in the Fig. 3.



**Fig. 3.** Incremental step. The constraints are assumed to be valid for each patch (top). When two patches are merged (bottom) the constraints need to be checked only for the new triangles (yellow).

Three kind of constraints are enforced:

- **Photo-Consistency Constraint:** the projections of a triangle on the images where it is visible should be photo-consistent.
- **Visibility Constraint:** a triangle must not to occlude any visible point.
- **Non Intersection Constraint:** a triangle must not intersect any previously defined surface.

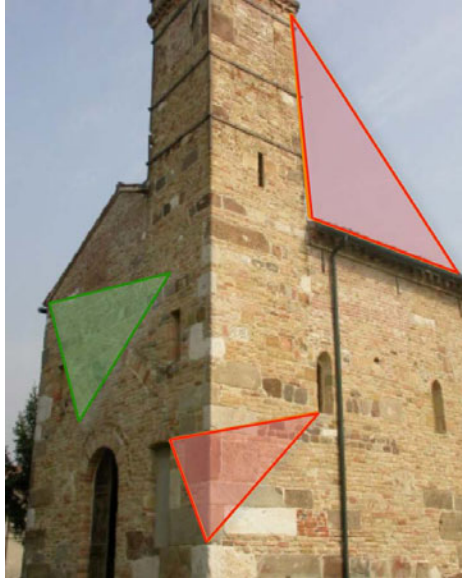
### 3.1 Photo-Consistency Constraint

A patch in space is *image-consistent* if all its projections onto the images where it is visible contain conjugate points. Image consistent patches are attached to actual object surfaces in the scene (see Fig. 4). Image-consistency can be checked through *photo-consistency*, the property that the projections of a patch are equal up to a projective transformation and photometric nuances.

Let us first define a set of compatible images as the ones where the vertices of a given triangle are visible. Among them, the one where the projected triangle exhibits the maximum area is chosen as the reference. All the triangles in the compatible images are projectively warped onto the triangle in the reference image and compared to it through normalized cross-correlation (NCC). The final photo-consistency of the 3D triangle is obtained as the average of the NCC scores of its projections (the value ranges from  $-1$  to  $1$ ), and it is considered photo-consistent if this value is below a fixed threshold.

### 3.2 Visibility Constraint

A Structure and Motion pipeline generally outputs the *visibility* of the points, i.e. the cameras from which a point is visible. This information can be exploited



**Fig. 4.** The green triangle is image-consistent, the red ones are not

to formulate a simple yet powerful constraint: a surface patch must not occlude a 3D point from the view where it is visible.

Mathematically, this translates into a segment-triangle intersection test. The segment ranges from the optical center of the view to the 3D point that is being examined. The intersection test can be performed efficiently at constant time. However, in the worst case - i.e. when no intersections with the current triangle were found - one needs to run the test for each view and for each visible point from that view. In order to speed up the process, we precompute the axis aligned bounding box (AABB) for each view that contains every visible point and the optical center. We also compute and update an AABB that contains every point of a patch. A prior intersection test is made between the AABB of the patch and the AABB of a view: if no intersection occurs we are assured that no triangle of the patch will intersect a segment in that view. The intersection test between two AABB also takes constant time.

### 3.3 Non Intersection Constraint

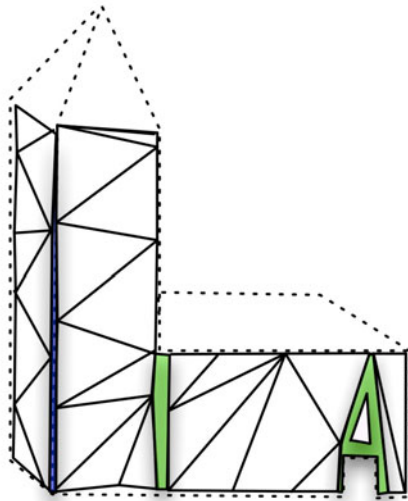
During the patch growing, it may happen that patches end up intersecting each other in their interior. This is clearly an unwanted situation, as the customary assumption holds that surfaces are manifolds. To avoid this, we embed the non intersection constraint directly in the J-linkage.

When creating a new patch we check that it is not intersecting any previously defined patch. This translates into a triangle-triangle intersection test among all the triangles of two patches. The triangle-triangle intersection test can be

computed in constant time. However, when dealing with surfaces composed by many triangles, it may require many checks. We speed up the process taking advantage of the AABB computed for every patch.

## 4 Filling the Gaps

During the agglomerative clustering of J-linkage, it is sufficient that a single triangle does not satisfy a constraint to discard the entire merge, because it is inductively assumed that patches are *convex*. As a result, triangles that fulfill the constraints are discarded, thereby leaving gaps in the surfaces between neighbouring patches (Fig. 5). This issue is solved *a-posteriori*, by a gap-filing heuristics that relaxes the convexity assumption.



**Fig. 5.** Green regions are gaps between adjacent patches that are to be filled. Blue regions are gaps between orthogonal patches.

Two patches are said to be *adjacent* if at least one of the points of one patch contains a point of the other patch in his  $k$ -neighborhood. We can distinguish two cases of adjacent patches: *coplanar*, when the angle between the respective support planes is less than 30 degrees, and *orthogonal*, when the angle lays between 60 and 120 degrees. A graph of connection between the patches can thus be inferred. First, we fill the gaps between orthogonal patches. By construction, a point can belong to only one patch. We identify the points compatible, by means of the inlier threshold, to both the orthogonal patches. The points are then added to both patches if the constraints defined before are valid for the newly computed patches.



Finally, we fill the gaps between coplanar patches by testing each one connecting triangles between the patches using the same methods and constraints defined before. When two patches have been processed they are treated as a single entity in this iterative procedure.

## 5 Results

We tested our method on real data coming from a completely automatic SaM pipeline. The first test compares our approach to [13] and [21] on a simple object. The second set of experiments demonstrate how our method can cope with real-world examples.

### 5.1 Comparative Test

In order to be able to tun a comparative test with [13] we had to choose a setup where all the points projects in all the views. To this end we constructed the “Duplo” object visible in Fig. 6 and manually selected 72 keypoints correspondences in 5 views. The 3D structure have been recovered by a SaM pipeline. We also considered for comparison the original J-Linkage without additional constraints and gap filling procedures. The results are shown in figure Fig. 6. It can be noticed how [13] fails to extract a consistent triangulation. The reason is that the simple subject of the scene is non-convex, and photo-consistency alone seems to be sufficiently powerful in this case. J-Linkage without constraints is able to correctly detect the supporting planes; yet, the final patches defined with a Delaunay triangulation contains gaps and fails to delineate the underlying object. Our approach obtains best results, even if some triangles are missing.

### 5.2 Real World Examples

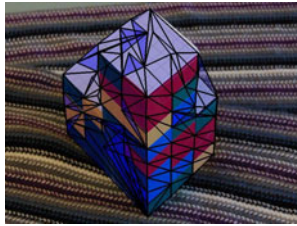
Three tests were performed on publicly available data<sup>1</sup> produced by the Structure and Motion pipeline described in [20]. Results show the fitting planes to the cloud of points, and the associated patches, projected over the images.

The first set - “Dante” - is composed of 39 images and 2971 points. The results are shown in Fig. 7. In the second test the subject is a church. The images involved are 54 and the cloud of points is composed of 11094 points. The results are shown in Fig. 8. The last test is computationally more challenging. The subject is “Piazza Bra” (Verona). The images are 380 and the points 52024 (obtained by subsampling the original 104047 points). The final extracted patches with our approach, visible in Fig. 9, are 302. It can be appraised from the examples shown as the patches are always covering planar regions of *actual* surfaces, whereas planes found by J-linkage not always correspond to a physical plane (see for example the triangle in the sky of Fig. 8(a)). Please note that the boundaries of the patches seldom do not coincide with the actual edges of the façades, because points were detected by SIFT, which tends to keep away from

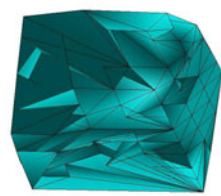
<sup>1</sup> <http://profs.sci.univr.it/~fusiello/demo/samantha/>



(a) Triangulation produced by [13].



(b) Triangulation produced by [13].



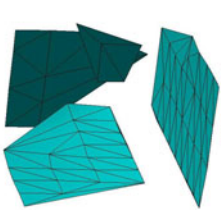
(c) Final surface produced by [13].



(d) Triangulation produced by J-Linkage.



(e) Triangulation produced by J-Linkage.



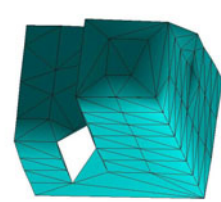
(f) Final surface produced by J-Linkage.



(g) Triangulation produced by our approach.

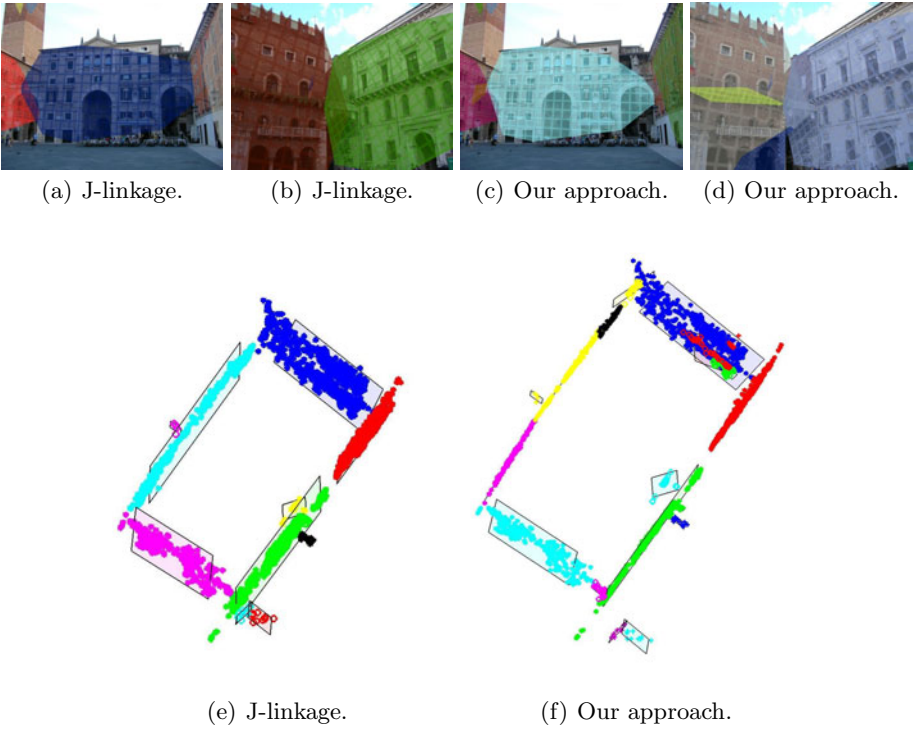


(h) Triangulation produced by our approach.

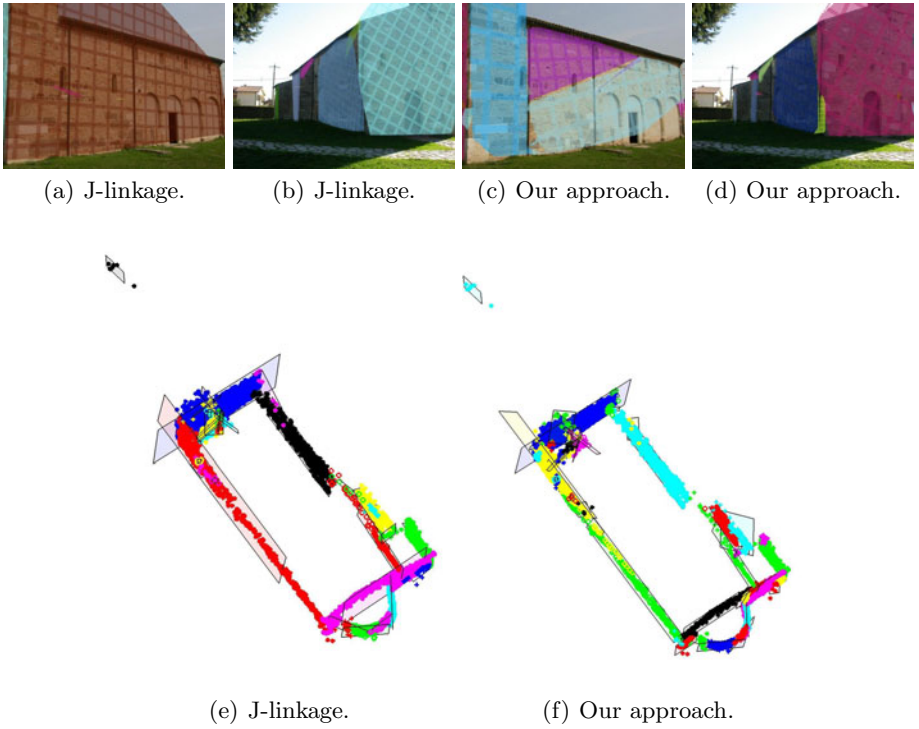


(i) Final surface produced by our approach.

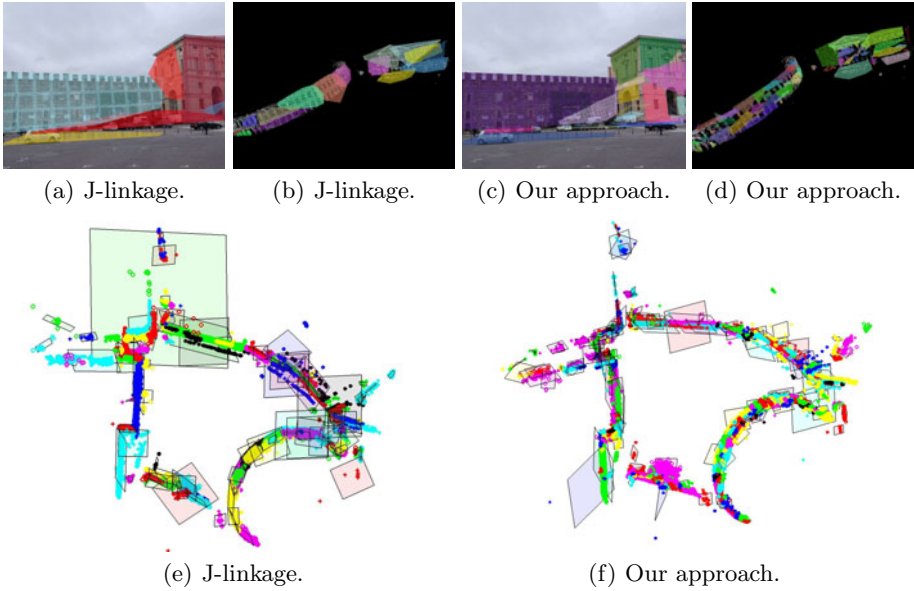
**Fig. 6.** “Duplo” example. The top row depicts the results produced by [13], the middle row the results produced by J-linkage followed by a Delaunay triangulation and the bottom row shows the results of our approach.



**Fig. 7.** “Dante” dataset. The top row (a-c) depicts the patches superimposed onto the images. The bottom row (e,f) shows the supporting planes from an azimuth view.



**Fig. 8.** “Pozzoveggiani” dataset. The top row (a-c) depicts the patches superimposed onto the images. The bottom row (e,f) shows the supporting planes from an azimuth view.



**Fig. 9.** “Piazza Bra” dataset. The top row (a-c) depicts the patches superimposed onto the images. The bottom row (e,f) shows the supporting planes from an azimuth view.



**Fig. 10.** Textured examples

corners. However, these planar patches must be considered only as a initial step toward the extraction of an high-level model. Several heuristics can be deployed to expand the regions up to their natural boundaries.

The code is entirely written in C++ and it is written upon J-Linkage<sup>2</sup>. The computing time on an entry level PC with a single core 2.4Ghz cpu, is about , 20 seconds for the “Duplo” example, 15 minutes for “Dante”, 1 hour for “Pozzoveggiani” and 14 hours for “Piazza Bra”.

For visualization purposes only we produced a textured version of our results shown in Fig. 10. The procedure we followed is straightforward: For every patch

<sup>2</sup> <http://profs.sci.univr.it/~fusiello> - <http://www.toldo.info/roberto>

we have defined an alpha blended textured quad. The quad coordinates are settled in order to include all the points of the patch projected on the supporting plane.

## 6 Discussion

In this work we proposed a novel method for extracting planar photo-consistent patches that can cope with fairly large and noisy datasets coming from a standard SaM pipeline.

The spatial information has been seamlessly combined with the information coming from the images and the SaM pipeline. The final result is a very compact and stable intermediate representation, and can be regarded as a starting point for a complete automatic reconstruction of scene surfaces. Future work will aim at bridging further the semantic gap.

## References

1. Taylor, C., Debevec, P., Malik, J.: Reconstructing polyhedral models of architectural scenes from photographs. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065, pp. 659–670. Springer, Heidelberg (1996)
2. Cipolla, R., Robertson, D., Boyer, E.: Photobuilder-3d models of architectural scenes from uncalibrated images. In: IEEE Int. Conf. on Multimedia Computing and Systems, vol. 1, pp. 25–31 (1999)
3. Wilczkowiak, M., Sturm, P., Boyer, E.: Using geometric constraints through parallelepipeds for calibration and 3D modeling. *IEEE transactions on pattern analysis and machine intelligence*, 194–207 (2005)
4. van den Hengel, A., Dick, A., Thormählen, T., Ward, B., Torr, P.: VideoTrace: rapid interactive scene modelling from video. In: Proceedings of the SIGGRAPH conference, vol. 26, ACM, New York (2007)
5. Schindler, K., Bauer, J.: A model-based method for building reconstruction. In: IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis, pp. 74–82 (2003)
6. Alegre, F., Dellaert, F.: A probabilistic approach to the semantic interpretation of building facades. Technical report, Georgia Institute of Technology (2004)
7. Dick, A., Torr, P., Cipolla, R.: Modelling and interpretation of architecture from several images. *International Journal of Computer Vision* 60, 111–134 (2004)
8. Brenner, C., Ripperda, N.: Extraction of facades using rjM-CMC and constraint equations. In: Photogrammetric Computer Vision, pp. 155–160 (2006)
9. Tian, Y., Zhuh, Q., Gerkea, M., Vosselmana, G.: Knowledge-Based Topological Reconstruction for Building Façade Surface Patches. In: 3D Virtual Reconstruction and Visualization of Complex Architectures (3D-ARCH), vol. 18 (2009)
10. Han, F., Zhu, S.: Bayesian reconstruction of 3d shapes and scenes from a single image. In: Workshop on High Level Knowledge in 3D Modeling and Motion, vol. 2 (2003)
11. Muller, P., Zeng, G., Wonka, P., Van Gool, L.: Image-based procedural modeling of facades. *ACM Transactions on Graphics* 26, 85 (2007)
12. Hilton, A.: Scene modelling from sparse 3D data. *Image and Vision Computing* 23(10), 900–920 (2005)

13. Morris, D., Kanade, T.: Image-consistent surface triangulation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1999), vol. 1. IEEE Computer Society, Los Alamitos (2000)
14. Nakatsuji, A., Sugaya, Y., Kanatani, K.: Optimizing a triangular mesh for shape reconstruction from images. *IEICE Transactions on Information and Systems*, 2269–2276 (2005)
15. Cooper, O., Campbell, N., Gibson, D.: Automatic augmentation and meshing of sparse 3D scene structure. In: Seventh IEEE Workshops on Application of Computer Vision, vol. 1 (2005)
16. Torr, P., Zisserman, A.: MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding* 78, 138–156 (2000)
17. Bartoli, A.: A random sampling strategy for piecewise planar scene segmentation. *Computer Vision and Image Understanding* 105, 42–59 (2007)
18. Moser, S., Wahl, R., Klein, R.: Out-of-Core Topologically constrained simplification for city modeling from digital surface models. In: *3D Virtual Reconstruction and Visualization of Complex Architectures (3D-ARCH)*, vol. 18 (2009)
19. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3D. In: *SIGGRAPH: International Conference on Computer Graphics and Interactive Techniques*, pp. 835–846 (2006)
20. Farenzena, M., Fusiello, A., Gherardi, R.: Structure-and-Motion Pipeline on a Hierarchical Cluster Tree. In: *3-D Digital Imaging and Modeling (3DIM)* (2009)
21. Toldo, R., Fusiello, A.: Robust multiple structures estimation with J-linkage. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part I. LNCS*, vol. 5302, pp. 537–547. Springer, Heidelberg (2008)
22. Furukawa, Y., Curless, B., Seitz, S., Szeliski, R., Szeliski, R.: Towards Internet-scale Multi-view Stereo. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 12 (2010)
23. Furukawa, Y., Curless, B., Seitz, S., Szeliski, R.: Reconstructing building interiors from images. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 80–87 (2009)
24. Myatt, D.R., Torr, P.H.S., Nasuto, S.J., Bishop, J.M., Craddock, R.: Napsac: High noise, high dimensional robust estimation - it's in the bag. In: *British Machine Vision Conference* (2002)

# Contour Grouping and Abstraction Using Simple Part Models

Pablo Sala and Sven Dickinson

Department of Computer Science, University of Toronto, Toronto ON, Canada

**Abstract.** We address the problem of contour-based perceptual grouping using a user-defined vocabulary of simple part models. We train a family of classifiers on the vocabulary, and apply them to a region oversegmentation of the input image to detect closed contours that are consistent with some shape in the vocabulary. Given such a set of consistent cycles, they are both abstracted and categorized through a novel application of an active shape model also trained on the vocabulary. From an image of a real object, our framework recovers the projections of the abstract surfaces that comprise an idealized model of the object. We evaluate our framework on a newly constructed dataset annotated with a set of ground truth abstract surfaces.

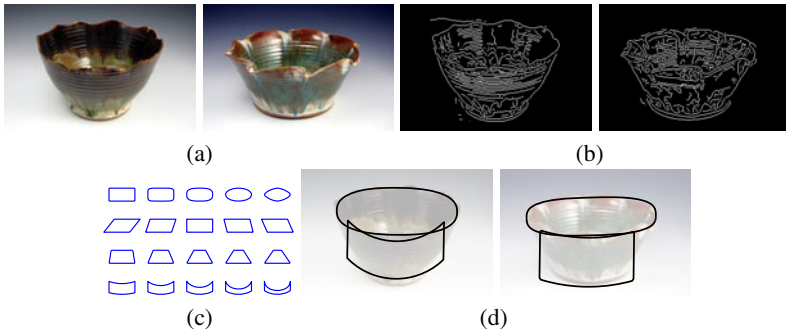
**Keywords:** perceptual grouping, shape abstraction, part vocabulary.

## 1 Introduction

The problem of computational perceptual grouping received considerable attention before the advent of appearance-based recognition, when object models were typically shape-based and image features were typically contour-based. Moreover, while object databases were rather small, it was generally assumed that a linear search of a database, i.e., matching the image features against each model in succession and choosing the best-matching model, was an unacceptable strategy, for it did not scale to very large databases. In an effort to achieve sublinear scaling, much effort was devoted to the problem of object *indexing*, i.e., using a set of image features to query the database for candidate objects that might account for the image features. An effective query structure, or index, should be small enough to be reliably extracted, yet discriminative enough to aggressively prune the database down to a few promising candidates. Since image features were contour-based, perceptual grouping played a major role in grouping together contours that were unlikely to co-occur by chance. Moreover, grouping was based *not* on object-level prior knowledge, but rather on mid-level (object-independent) prior knowledge. Such grouping was essential, since local contour features were highly ambiguous, and without grouping them into more discriminative structures, effective indexing into large databases was problematic.

The object categorization community's focus on the object *detection* problem has since drawn attention away from perceptual grouping, since there is no need to construct an effective index when the candidate (target) object is known. However, there are signs that the categorization community is not only returning to the more categorical feature of object shape, but to the more general problem of recognition from a large





**Fig. 1.** Recovering abstract shape parts from an image: (a) input image of two exemplars that show considerable within-class variation; (b) extracted contours: note that corresponding contour-based features are seldom in one-to-one correspondence; (c) a simple example vocabulary of 2-D part models that will drive the perceptual grouping and shape abstraction processes; (d) the resulting abstract surfaces recovered by our framework; contour correspondence exists not at the level of individual contours, but at a much higher level of abstraction.

database. In turn, the need to group together contour features into powerful indexing structures may stimulate interest in perceptual grouping [1–3]. However, whereas simple groups of contour features may have been sufficient for indexing into a database of shape exemplars, today’s interest in categorization will require not only the grouping of causally related contour features, but their *perceptual abstraction* to yield higher-order shape features that are invariant to within-class variation. This raises two important challenges: 1) how to perceptually group related contours; and 2) how to perceptually abstract the groups into high-order shape features that are more generic and less specific.

In this paper, we present a novel approach to the perceptual grouping and abstraction of image contours using a set of 2-D part models. We assume no object-level prior knowledge and, like the perceptual grouping community, assume only a mid-level shape prior. However, our shape prior is slightly stronger than such classical Gestalt features as symmetry, parallelism, collinearity, etc. Specifically, our mid-level shape prior takes the form of a user-defined vocabulary of simple 2-D shape models, representing a fixed set of parts from which a large database of object models can be constructed. In that sense, our vocabulary can be seen as a high-level nonaccidental regularity – a common denominator set of part shapes that can be used to model a large collection of objects in the world. Since different domains may demand different vocabularies of parts, it’s essential that our framework be *independent* of the part vocabulary; therefore, the vocabulary is an input to our framework.

Figure 1(a) shows images of two object exemplars that belong to the same class (bowl), while Figure 1(b) shows their extracted contours; note that corresponding contour-based features are seldom in one-to-one correspondence. In Figure 1(c), we show sample instances from a simple vocabulary of 2-D shapes that will be used to group and abstract the contours in Figure 1(b). In Figure 1(d), we overlay the abstract shapes recovered by our algorithm. If we examine carefully the region boundaries in both images, we observe that due to within-class variation or noise, there are few

corresponding contours between the two parts. As noise and within-class variation increase, methods that rely on one-to-one feature correspondences among specific contour-based features may fail. Only by examining the abstract shapes defined by these contours does commonality between the two exemplars emerge.

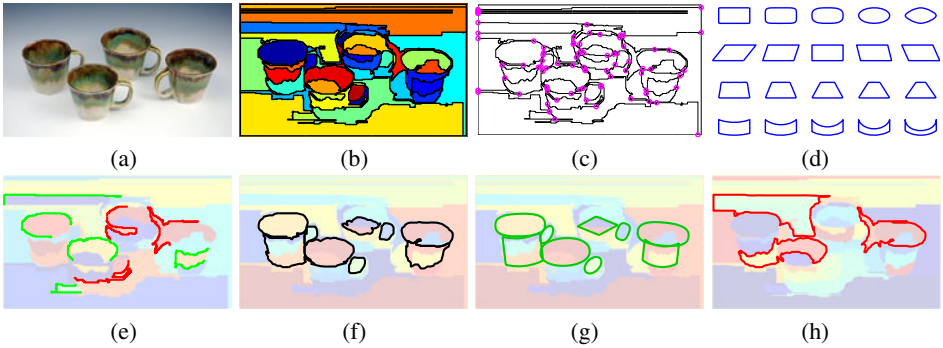
## 2 Related Work

There is a large body of work on using simple shape models to group and regularize 2-D contour data. Due to space constraints, we will not review approaches that take, as input, a silhouette, i.e., assume figure-ground segmentation, nor will we review approaches that assume knowledge of what object is in the scene, i.e., object-level shape priors. Rather, we adopt the classical perceptual grouping position and review related approaches that assume only mid-level shape priors. Such priors can range from simple smoothness to compactness to convexity to symmetry to more elaborate part models, but stop short of object models.

Jacobs [4] and Estrada and Jepson [5] explored the nonaccidental regularity of convexity to group contours into convex parts. Other researchers, e.g., Stahl and Wang [3], have explored the nonaccidental regularity of symmetry to group contours into symmetric parts, while Lindeberg [6], has explored symmetry to extract symmetric blobs and ridges directly from image data. While each of these models exploits a particular nonaccidental shape regularity, they also restrict the image domain. Moreover, each mid-level shape prior comes with its own computational model, and there is little to unify the approaches.

More powerful part models stemmed from the early recognition by parts paradigm. Pentland [7] partitioned a binary image into 2-D parts corresponding to the projections of a vocabulary of 3-D deformable superquadrics. The method focused more on the problem of part selection (from a large space of part hypotheses) than the grouping of features into parts, and the framework was never applied to contours. Pilu and Fisher [8] attempted to recover 2-D deformable parts models from image contours. However, they assumed that the correspondence between image and model contours was one-to-one, restricting the scenes to contain very simple objects. Little abstraction was achieved, and such systems were rarely applied to complex scenes.

The dual problem to fitting part models to contours is fitting part models to regions. Liu and Sclaroff [9] proposed a method capable of finding instances of a 2-D shape (possibly a part model) in an image. From a bottom-up region segmentation, the space of region merges and splits is explored in search of region groups with shapes similar to a 2-D statistical template model. Wang et al. [2] proposed a stochastic approach to explore the space of region merges and splits in search of region groups having a particular shape. From a bottom-up image region segmentation, their approach was capable of finding multiple occluded instances of a model shape by grouping oversegmented regions. However, these approaches typically admit a single model shape and also rely strongly on appearance homogeneity to guide the grouping process. Moreover, Wang et al.'s method employed a detailed model of the shape, and did not attempt shape abstraction. In [10], we introduced a model-based framework to detect abstract part hypotheses from a multiscale edge map. However, it was not only computationally expensive (since



**Fig. 2.** Problem Formulation: (a) input image; (b) region oversegmentation; (c) region boundary graph; (d) example vocabulary of shape models (used in our experiments); (e) example paths through the region boundary graph that are consistent (green) and inconsistent (red); (f) example detected cycles that are consistent with some model in the vocabulary; (g) abstractions of cycles consistent with some model; (h) example cycles inconsistent with all models.

it exhaustively explored all possible transformations of every model), it also generated a large number of hypotheses (several hundred), thus yielding poor precision.

In summary, a diverse set of mid-level shape priors have been proposed, each with its own strengths and weaknesses. Unfortunately, most approaches are closely tied to their underlying shape priors, and the mechanism for recovering one class of parts may vary greatly from the mechanism for recovering the next class. Moreover, most part recovery schemes are rather brittle and offer little opportunity for recovering abstract parts from the noisy, irregular contours that often make up real objects. We address these two shortcomings head on with a part-based grouping framework that's independent of the parts, and a shape abstraction mechanism that can recover the abstract parts that make up a large collection of real objects.

### 3 Overview of the Approach

Our approach begins by computing a region oversegmentation (Figure 2(b)) of the input image (Figure 2(a)). The resulting region boundaries yield a *region boundary graph* (Figure 2(c)), in which nodes represent region boundary junctions where three or more regions meet, and edges represent the region boundaries between nodes; the region boundary graph is a multigraph, since there may be multiple edges between two nodes. Our approach can be formulated as finding simple cycles in the region boundary graph whose shape is consistent with one of the model shapes in the input vocabulary (Figure 2(d)); these are called *consistent cycles*. There is an exponential number of simple cycles in a planar graph [11], and simply enumerating all cycles (e.g., [12]) and comparing their shapes to the model shapes is intractable. Instead, we start from an initial set of starting edges and extend these paths, called *consistent paths* (or CPs), as long as their shapes are consistent with a part of *some* model. To determine whether a given path is consistent (and therefore extendable), we approximate the path at multiple scales with

a set of polylines (piecewise linear approximations, providing a set of low-dimensional boundary abstractions), and classify each polyline using a one-class classifier trained on the set of training shapes (Figure 2(e)). When a consistent path is also a simple cycle, it is added to the set of output consistent cycles (Figure 2(f)).

Figure 2(d) shows the input vocabulary used in our experiments: four part classes (supere ellipses plus sheared, tapered, and bent rectangles, representing the rows) along with a few examples of their many within-class deformations (representing the columns). It is important to note that our approach is independent of the vocabulary of parts. While our demonstration vocabulary is ideally suited to the projected surfaces of ellipsoids, and straight and bent rectangular blocks and cylinders, representing a simple volumetric part vocabulary, the approach can accommodate any set of shapes, parameterized or otherwise. Each shape model is allowed to anisotropically scale in the  $x$ - and  $y$ -directions as well as rotate in the image plane. Since we employ scale-, rotation-, and translation-invariant features to train the classifiers, we need to generate only (approx.) 1,500 instances (by varying the aspect ratio and deformation parameters) belonging to these four shape classes. A *single* classifier is trained on all the component polylines of length (i.e., number of piecewise linear segments)  $k$  spanning the *complete* set of shape models and their deformations. Therefore, if  $K$  is the upper bound on the length of a polyline approximating a shape in the vocabulary, then  $K$  classifiers are trained. An ideal vocabulary defines a small set of “building blocks” common to a large database of objects. As such, the complexity of the vocabulary shapes is low, and even at the finest scale of polyline partitioning of a vocabulary shape’s contour,  $K$  remains low; for our vocabulary,  $K$  is 13.

The algorithm outputs cycles of contours that are consistent with one of the model (training) shapes. A cycle consists of actual contours (edges in the region boundary graph) in the image, and therefore does not explicitly capture the abstract shape of the contours. Moreover, the cycle has not yet been categorized according to the shapes in the vocabulary. To abstract (or regularize) the shape of a cycle and to categorize it, we follow a standard, iterative 2-step active shape model (ASM) fitting framework [13] trained on about 600,000 model instances, generated by varying their aspect ratio, orientation, and a finer sweeping of the deformation parameters than the one used to train the polyline classifiers. We iterate over the classical two-step ASM procedure, consecutively aligning and deforming the mean training shape with the cycle until convergence. However, we depart from a standard ASM framework in two key ways.

In a standard ASM framework, the training shapes belong to a single shape class, and the allowable, often limited, deformations are typically captured (using PCA) in a low-dimensional shape space that can be approximated by a multidimensional Gaussian distribution. Moreover, at run time, the model must be properly initialized, for if the model is grossly misaligned, the deformations required to warp the model into the image may fall outside the space of allowable deformations. In our case, given a consistent cycle, we don’t know which category of vocabulary shape it belongs to, and hence which ASM model to apply (if we assumed one model per category in the vocabulary). Moreover, even if we knew its category, we assume no correct or near-correct initial landmark correspondence. We overcome the first problem by having a single ASM that’s trained on all instances of all the shapes in the vocabulary, and overcome

the second problem by training on all possible landmark correspondences (alignments) across these shapes.

After ASM convergence, the training shape closest to the deformed model identifies the category of the cycle. In the previous step, the consistent cycle classifier's precision rate is never 100% at reasonable recall rates, and some of the recovered consistent cycles (of contours) may yield shapes that are qualitatively different from those in the vocabulary. Therefore, following ASM convergence, shapes that are still significantly different from the training shapes are discarded. Figure 2(g) illustrates the vocabulary shapes abstracted from the consistent cycles in Figure 2(f); for each detected shape, the algorithm also yields its shape category. Finally, Figure 2(h) illustrates some of the false positives discarded by the shape abstraction process.

## 4 Finding Consistent Cycles

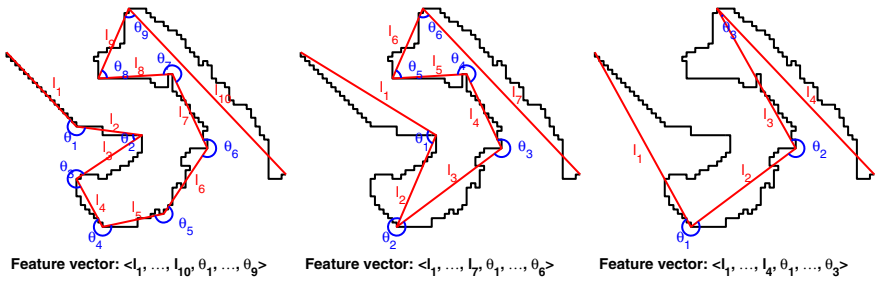
In the following subsections, we elaborate on the steps of our algorithm for finding consistent cycles, i.e., cycles whose shape is consistent with one of the model shapes; Section 5 will focus on the problem of abstracting/categorizing the shape of the cycle.

### 4.1 Path Initialization

The goal of path initialization is to identify a minimum cardinality set of edges such that every cycle in the graph contains at least one of the edges. This can be easily achieved by computing the *feedback edge set*, i.e., the smallest set of edges whose deletion results in an acyclic graph. The feedback edge set is computed as the edge complement of a spanning tree. Favoring edges that represent longer, and thus more informative contours, we will choose as our initial edge set the edge complement of the minimum spanning tree, where edge weight equals contour length. While every cycle contains at least one of these initial edges, two or more of these edges may be part of the same cycle. This is problematic, since extending these initial paths will yield the same cycles, which is highly inefficient. We avoid this problem by imposing a total ordering on the edges, and allowing a path to be extended only by an edge whose rank is greater than that of the initial edge of that path; the edges in the minimum spanning tree are all assigned a rank of  $\infty$ . The rank of a path is the rank of its initial edge. The set of initial edges, and their ranks, form our initial set of paths, and they are added to the queue of paths to be extended.

### 4.2 Path Extension

At each iteration of the algorithm, one of the paths is taken off the queue. If the path is a cycle, its consistency with the vocabulary of model shapes is checked. If it's consistent with at least one shape in the vocabulary, it is added to the output list of consistent cycles. If, however, the path is not a cycle, its consistency is also checked. If the path is consistent with a portion of the boundary of at least one shape in the vocabulary, then the path's possible extensions by an edge whose rank is greater than the path are added to the queue. The algorithm continues until the queue is empty, and outputs the consistent cycles.



**Fig. 3.** Feature vectors for a multiscale polyline approximation of a contour

Checking consistency begins by approximating the shape of the cycle or path with a polyline computed at different scales using the Ramer-Douglas-Peucker algorithm [14]. For each resulting polyline, we compute a feature vector that encodes the angles and normalized lengths of the linear segments making up the polyline. As illustrated in Figure 3, the length of the feature vector is a function of the number of linear segments comprising the polyline. Each feature vector is passed to a one-class classifier (there is a classifier for each feature vector length) that determines if the feature vector is geometrically close to one of the training feature vectors. Those scales at which their corresponding polylines are consistent are associated with the path. If a path at a particular scale is not consistent, then no extension of that path can be consistent at that scale. Thus, when a path is initialized, it is associated with all scales, and when it is extended, its associated scales can only remain constant or decrease. If there is no scale at which the path is consistent, the path is discarded, and will not be extended further.

### 4.3 Training the Classifiers

The feature vectors used to train the classifiers are generated from contour fragments of model instances. Axis-aligned instances of within-class deformations of each model are generated at varying aspect-ratios, and Gaussian noise is added to each generated contour with a standard deviation proportional to the model size (defined as the average distance from a model contour point to the model's centroid). A number of equidistant points along each generated contour are sampled, and the two subcontours between every pair of such points (traversed in both directions) are used as a training example. A feature vector is generated for each subcontour from its polyline approximation, computed using a tolerance that is proportional to model size. Finally, the dimensionality of the feature vectors is reduced using PCA. Classification is performed on these reduced dimensionality vectors. For our model vocabulary, we observe that at least 99% of the variance is, in general, captured by the top  $N$  PCA components for the case of feature vectors of dimension  $2N - 1$ , corresponding to polylines with  $N$  linear segments.

In our implementation, the number of linear segments comprising the longest polyline approximating a model's contour is bounded by 13. For this reason, at consistency check time, a path whose polyline approximation is longer than this value is discarded as inconsistent. Moreover, in the case of the path being a cycle, if its polyline

approximation at a certain scale has less than three linear segments, the cycle is deemed inconsistent at that scale, since the scale is obviously too coarse. In the case of an open path with a polyline approximation with less than two segments, there is not enough information to decide on the path’s consistency, since the path is evidently still too short at the given scale. In this case, the path is treated in the algorithm as if it had been consistent, leaving the decision to a future iteration, after it has been eventually extended to a length where a consistency decision is possible.

For our experiments, approximately 4 million contour fragments were employed to train the classifiers. Due to the difficulty in generating an adequate set of training examples of inconsistent model contour fragments, a one-class classifier was used instead of a binary classifier. Since the consistency check needs to be performed a large number of times (once per path extension), an efficient implementation calls for a method with a low classification complexity for this task. We obtain good classification performance and very fast classification rates using a Nearest Neighbor Data Description approach [15] implemented via a fast approximate nearest neighbor search data structure [16].

## 5 Abstracting the Shape of a Consistent Cycle

As mentioned in Section 3, we employ an ASM to both abstract the shape of a consistent cycle and to categorize it. Recall that we train a single ASM on all deformations of all vocabulary shape classes over all possible landmark correspondences. This avoids a proliferation of ASM models (one model, regardless of the size of the vocabulary), and allows the model to be initialized anywhere on the cycle. To train the Point Distribution Model, we sweep the parameter space of the shapes in the model vocabulary. Specifically, we generate contours for all models and degrees of deformation (i.e., bending, tapering, and shearing) at a dense set of discrete aspect ratios and all possible cyclical landmark alignments; in our implementation, we generate a total of approximately 600,000 such training cases. Rotation invariance is achieved by training on all model landmark alignments corresponding to all possible  $N$  cyclical rotations of the landmarks, while scale invariance is achieved during ASM fitting by the rigid transformation estimation.

The list of landmark points  $\mathbf{m}_1, \dots, \mathbf{m}_N \in \mathbb{R}^2$  in a training example corresponds to a fixed number of equidistant points sampled along the contour ( $N = 64$  in our experiments). The first landmark  $\mathbf{m}_1$  is the one for which the vector from the centroid  $\bar{\mathbf{m}}$  to the landmark has the lowest (absolute value) angle with respect to the x-axis, i.e.,  $(1, 0)^T \cdot (\mathbf{m}_i - \bar{\mathbf{m}}) / \|\mathbf{m}_i - \bar{\mathbf{m}}\|$  is maximum for  $i = 1$ ; in case of a tie, we choose the point with maximum  $\|\mathbf{m}_i - \bar{\mathbf{m}}\|$ . The indices of the other landmarks in the list keep their natural order along the contour. Following the standard ASM approach, a vector is formed for each contour by rasterizing the contour’s landmark coordinates. A PCA basis is computed for the training set, and the lowest-order principal components that capture most of the variance are chosen. For our training set, 99.9% of the total variance is captured by the top 21 components.

Fitting is performed by the successive iteration of two steps: one that finds the rigid transformation that best aligns the current deformed model to the query contour, and a second step that adjusts the shape parameters that deform the model to better improve

the fit. This adjustment is constrained such that the deformed model shape is consistent with that of the training examples. Enforcing this constraint is accomplished by checking that the adjusted shape parameters do not fall outside of the distribution of the training shapes; if the parameters do fall outside, they are set according to the closest shape in the distribution.

We differ from a standard ASM in how a shape adjustment is constrained to lie in the space of training shapes. In a classical ASM framework, where training shapes belong to the same class and exhibit relatively minor deformations, the low-dimensional subspace where the shape points live is approximately Gaussian distributed. Under this condition, enforcing this constraint amounts to simply verifying that the adjusted shape parameters do not deviate more than a certain number of standard deviations (e.g., 3) from the mean shape; beyond that, the point is scaled down to correspond to the closest point within the distribution. In our case, the set of training shapes is quite heterogeneous (spanning multiple categories and all possible initial correspondences), yielding a complex shape space boundary whose enclosed distribution is not well approximated by a Gaussian. We therefore need a different way to constrain a given adjusted shape to fall within the shape space spanned by the training set. Since our training set densely covers the space of shapes of interest and a low-dimensional subspace captures most of the shape variance, a Nearest Neighbor Data Description method [15] provides a fast mechanism for checking if a query shape belongs to the target distribution. If an adjusted shape does not belong to the distribution, it is constrained to be in the distribution by replacing it by a near neighbor in the distribution. To avoid falling in local minima, we randomly choose the replacement from among the  $k(t)$  nearest neighbors, where  $k(t) \in \mathbb{N}_{>0}$  is a non-increasing function of the number of iterations. In our experiments, we obtained good results by using a linear function for  $k(t)$ .

Since we attempt to bridge the gap between image contours and ideal model contours, a simple distance field between image and model contour landmarks is inappropriate as a driving force to guide the model deformation process. Such an approach would give the same weight to all contour landmark correspondences and may fail to deform the model appropriately in the case of minor region undersegmentation or minor contour shape deviation from the model. In order to cope with these conditions, we define the deformation force for each landmark  $i$  as:

$$\delta_i = (1 - \alpha(t))(q_i - m_i) + \alpha(t)(\text{closest}_q(m_i) - m_i), \quad (1)$$

where  $q_1, \dots, q_N \in \mathbb{R}^2$  are image contour landmarks sampled at equidistant positions along the contour,  $\text{closest}_q(m_i)$  is the point along the image contour  $q$  that is closest to model landmark  $m_i$ , and  $\alpha(t) \in [0, 1]$  is a bijective monotonically increasing function of the iteration number  $t$ . In this way, at the beginning of the fitting process, the attraction forces between the image and model contours are globally driven purely by landmark correspondences. This roughly aligns the model to the cycle. As the iterations proceed, the model deformation becomes increasingly driven by local contour attraction forces, giving more weight to the consensus of the image contours that are closest to the model, and thus letting the deformation process overlook significant image contour departure from the abstract model as well as some undersegmentation. Note that our fitting problem is more constrained than a standard ASM framework, since all landmark correspondences between a consistent cycle and the model ASM are known, and



the influence of outlier landmarks on a consistent cycle (and their resulting incorrect correspondences) is decreased over time (iterations). Thus, the ASM is initially fit to an *entire* closed contour (not just a portion), but it converges to fit the relevant portion.

Finally, it is possible that inconsistent cycles are misclassified as consistent. After convergence, if the distance between the cycle and the model exceeds a threshold, or the cycle coverage by the model (i.e., proportion of cycle contour covered by model landmarks) is poor, the cycle is discarded as a false positive. We compute the scale-independent distance

$$d(\mathbf{q}, \mathbf{m}) = \frac{1}{N} \sum_{i=1}^N \frac{\|\delta_i\|}{\sqrt{\frac{1}{N} \sum_{i=1}^N \|\mathbf{q}_i - \bar{\mathbf{q}}\|^2}} \quad (2)$$

between the fitted model  $\mathbf{m}$  and the cycle  $\mathbf{q}$ , where the denominator is a normalizing factor corresponding to the geometric mean distance between cycle landmarks and their centroid, thus making  $d(\mathbf{q}, \mathbf{m})$  a scale-independent measure with an intuitive geometric interpretation; in our experiments, we obtained good results using a threshold of 0.15.

## 6 Results

Unfortunately, we know of no benchmark dataset for evaluating part-based shape abstraction, nor are we aware of competing approaches for part-based shape abstraction using a vocabulary of simple part models, except for [10]. Therefore, in order to evaluate our framework, we created an annotated dataset with 67 images<sup>1</sup> containing object exemplars whose 3-D shape can be qualitatively described by cylinders and bent or tapered cubic prisms. The abstract visible surfaces of each 3-D shape were hand-labeled using 2-D models drawn from our vocabulary. Figure 4 illustrates the output of our system on a number of examples in the dataset: column (a) shows the input image; column (b) shows the region oversegmentation used as input to our algorithm, computed using the local variation approach by Felzenszwalb and Huttenlocher [17] with a fixed parameterization on all images; column (c) shows the consistent cycles from which the shapes in column (d) were abstracted, representing the recovered parts closest to the ground truth in column (e). The numbers inside recovered abstract parts in column (d) indicate the rank of the part among all recovered parts in that image, computed as a function of the distance to the contours of the cycles that they are abstracting. The target regions can sometimes rank low if their degree of abstraction is high compared to non-target regions in the image (whether real or segmentation artifacts) that require less abstraction. Note that in some cases, e.g., the blender body in row 8, the ideal ground truth part (e.g., corresponding to the projection of the body of a tapered cylinder) did not exist in the vocabulary.

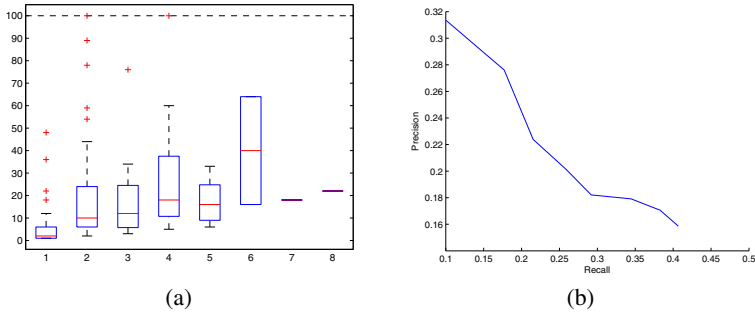
Our ability to abstract the shape of a cycle of contours with high local irregularity (shape “noise”) means that many false positive parts will be recovered. As a result, the ranks of some of the ground truth shapes among the hypotheses is poor. This is entirely due to the naive scoring mechanism (absolute fitting error) that tends to favor small,

<sup>1</sup> Available at <http://www.cs.toronto.edu/~psala/datasets.html>



**Fig. 4.** Abstract Part Recovery (see text for discussion)

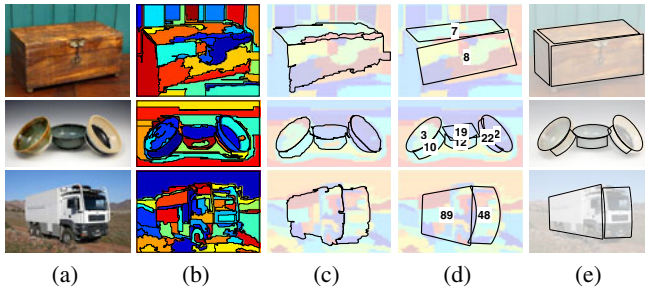
well-fitting shapes over larger abstractions. While more inspired scoring functions may increase the ranks of the target shapes, we mean only to illustrate the significant extent to which the target shapes are indeed generated. It is at a later stage, when contextual constraints are added, where we expect an aggressive pruning of false positives.



**Fig. 5.** Quantitative Evaluation: (a) Distribution of Rankings of Ground Truth Parts Among Generated Abstract Shapes. For example, the first detected ground truth part (leftmost column) in an image ranked 2nd (median rank - red bar) among the ranked hypothesis in the image. The top and bottom of the blue box defines the upper and lower quartiles, the whiskers define the furthest datum within 1.5IQR of the lower and upper quartiles, and the red “+”s the outliers. (b) Precision-recall curve (see text for discussion).

In future work, we plan to explore powerful contextual relations, including proximity, alignment, and 3-D shape information to prune many of these false positives. For example, if the surfaces in our images can indeed be the projections of volumetric parts, such as cylinders or prisms, then there are strong constraints on the shapes and relations of the component faces (parts) of their aspects. Other constraints are also possible, such as pruning smaller surfaces that are subsumed by larger surfaces. Adding these relational constraints is beyond the scope of this paper, and here we focus only on the initial recovery of the primitive parts. As can be seen from Figure 4, our framework is able to recover and abstract many of the surfaces of the objects.

To provide a quantitative evaluation of our framework’s ability to recover the correct abstractions amid the abstract shapes (hypotheses) recovered from an image, we analyze the rank of a ground truth shape among the ranked hypotheses. Figure 5(a) shows the distribution of ground truth part rankings. For example, the first detected ground truth part (leftmost column) in an image ranked 2nd (median rank - red bar) among the ranked hypothesis in the image. The second detected ground truth part (second column) ranked 10th, and so forth. The number of ground truth parts, on average, was 3, while the number of part hypotheses generated for an image, on average, was 71. No attempt was made to eliminate redundant models (i.e., models of the same category with roughly the same parameterization), and no size filtering was performed. From these results, we conclude that the target (ground truth) shapes reside in a manageable number of hypotheses, and we expect that with the application of contextual constraints, the false positive shapes can be drastically reduced. Figure 5(b) illustrates precision and recall for our database. While precision is low due to the high number of false positives (due to lack of contextual pruning and non-maximum suppression), our recall of ground truth shapes is reasonably good, typically failing in the presence of significant region undersegmentation. In terms of running time, a typical run of the consistent cycle detection algorithm requires an average of about 40,000 iterations, which takes about 3 seconds in our MATLAB/C++ implementation running on a laptop. The model



**Fig. 6.** Limitations of the Approach (see text for discussion)

abstraction algorithm was fully implemented in MATLAB and takes about 150 seconds to process all consistent cycles detected in an image.

Exploring the results in more detail, we see that Figure 4(d) shows the ability of our approach to abstract object surfaces that are locally highly irregular due to noise or within-class variation, but capture a model shape at a higher level of abstraction. In some cases (e.g., rows 5,6, and 8), we see misalignment with a neighboring shape. This can be due to two reasons: (1) the vocabulary may not contain the appropriate shape to model the surface; and (2) the shapes are recovered independently, with no alignment constraints exploited; such constraints, as well as other constraints, will play an aggressive role in pruning/aligning hypotheses in our future work. In all the examples, we can see that the model abstraction process is able to cope with region undersegmentation when it is restricted to a relatively small section of the contour. Figure 4 (rows 1 and 9) shows examples of cases in which, although some portions of the correct surface boundaries are missing, the models are still correctly fit due to the consensus of the correct surface contours.

Figure 6 illustrates some weaknesses of our approach. The top row shows a case in which an object’s surface is missing (i.e., box’s left face) due to strong undersegmentation of the input to our algorithm. Since the consistent cycle detection mechanism already keeps incremental hypotheses of partial contour matchings, in future work we plan to allow informative consistent paths to be abstracted (using a similar framework). This will not only accommodate region undersegmentation, but also region occlusion and partial part abstraction. In the second row of Figure 6, we see a case in which a consistent cycle was abstracted by a model of an incorrect category (i.e., the rim of the central bowl). This is because either the abstraction approach was trapped in a local minimum or there is an inherent shape ambiguity in the noisy contour. This can be remedied by allowing the abstraction process to return not just one model, but a list of candidate models that lie within a certain distance from the consistent cycle. We expect our future use of relational constraints to help overcome such ambiguity, and in this case “flip” the rectangle to an ellipse. Finally, the third row of Figure 6 shows a case where the ranking of the correct models is poor due to the presence of many uninteresting image region groups whose shape is consistent with vocabulary model shapes (i.e., there is a large number of region groups forming regular quadrilaterals). The use of context or non-maximum suppression can eliminate many of these false positives.

## 7 Conclusions

We have presented a framework for grouping contours (region boundaries) into parts according to a user-defined vocabulary of abstract parts models. Our contributions are threefold: (1) we train a classifier on all possible component fragments of a vocabulary of parts, and use the resulting set of classifiers to guide a grouping process that searches for cycles of locally irregular contours that are consistent, at some level of abstraction, with some model shape; (2) the consistent cycles are abstracted and categorized using a novel application of an ASM model which captures the entire vocabulary of shapes with a single model and which needs no proper initialization; (3) the resulting framework reports promising first steps toward part-based shape abstraction from images of real objects, and establishes a number of important directions for future work.

## References


1. Zhu, Q., Song, G., Shi, J.: Untangling cycles for contour grouping. In: ICCV (2007)
2. Wang, J., Gu, E., Betke, M.: Mosaicshape: Stochastic region grouping with shape prior. In: CVPR (2005)
3. Stahl, J., Wang, S.: Globally optimal grouping for symmetric boundaries. In: CVPR (2006)
4. Jacobs, D.W.: Robust and efficient detection of salient convex groups. PAMI 18, 23–37 (1996)
5. Estrada, F., Jepson, A.: Perceptual grouping for contour extraction. In: ICPR (2002)
6. Lindeberg, T.: Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. IJCV 11, 283–318 (1993)
7. Pentland, A.P.: Automatic extraction of deformable part models. IJCV 4, 107–126 (1990)
8. Pilu, M., Fisher, R.: Model-driven grouping and recognition of generic object parts from single images. In: ISIRS, Lisbon, Portugal (1996)
9. Liu, L., Sclaroff, S.: Deformable model-guided region split and merge of image regions. IVC 22, 343–354 (2004)
10. Sala, P., Dickinson, S.: Model-based perceptual grouping and shape abstraction. In: POCV, Anchorage, Alaska, pp. 1–8 (2008)
11. Buchin, K., Knauer, C., Kriegel, K., Schulz, A., Seidel, R.: On the number of cycles in planar graphs. In: Lin, G. (ed.) COCOON 2007. LNCS, vol. 4598, pp. 97–107. Springer, Heidelberg (2007)
12. Tiernan, J.C.: An efficient search algorithm to find the elementary circuits of a graph. Commun. ACM 13, 722–726 (1970)
13. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models - their training and application. CVIU 61, 38–59 (1995)
14. Douglas, D., Peucker, T.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. CC 10, 112–122 (1973)
15. Tax, D., Duin, R.: Data description in subspaces. In: ICPR, vol. 2, pp. 672–675 (2000)
16. Mount, D.M., Arya, S.: ANN: A library for approximate nearest neighbor searching (2006), <http://www.cs.umd.edu/~mount/ANN/>
17. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV 59, 167–181 (2004)

# Dynamic Color Flow: A Motion-Adaptive Color Model for Object Segmentation in Video

Xue Bai<sup>1</sup>, Jue Wang<sup>2</sup>, and Guillermo Sapiro<sup>1</sup>



<sup>1</sup> University of Minnesota, Minneapolis, MN 55455, USA

<sup>2</sup> Adobe Systems, Seattle, WA 98103, USA

**Abstract.** Accurately modeling object colors, and features in general, plays a critical role in video segmentation and analysis. Commonly used color models, such as global Gaussian mixtures, localized Gaussian mixtures, and pixel-wise adaptive ones, often fail to accurately represent the object appearance in complicated scenes, thereby leading to segmentation errors. We introduce a new color model, *Dynamic Color Flow*, which unlike previous approaches, incorporates motion estimation into color modeling in a probabilistic framework, and adaptively changes model parameters to match the local properties of the motion. The proposed model accurately and reliably describes changes in the scene’s appearance caused by motion across frames. We show how to apply this color model to both foreground and background layers in a balanced way for efficient object segmentation in video. Experimental results show that when compared with previous approaches, our model provides more accurate foreground and background estimations, leading to more efficient video object cutout systems. 

## 1 Introduction

Creating accurate masks for video objects is a fundamental component in the professional video post-processing pipeline. Once being accurately segmented from the video, the target objects can be used to create seamless composites, or be manipulated to create special visual effects. Recently, interactive, or user-guided video segmentation systems have gained considerable attention, given the fact that interactive systems can generate more accurate segmentation results than fully automatic ones, on a wide range of videos.

Although significant breakthroughs have been achieved in recent years on interactive video segmentation and matting  [\[1\]](#), this problem remains difficult for complex real world video sequences. The difficulty comes from two main aspects, namely *appearance complexity* and *motion complexity*. Appearance complexity refers to the fact that the targeted object could contain very similar, or even the same colors and features as the background, thus distinguishing the object from its background using color information becomes a hard problem.  [\[2\]](#) In addition,

---

<sup>1</sup> Work partially supported by NSF, NGA, ONR, ARO, and NSSEFF.

<sup>2</sup> Not limited to colors, the object appearance can also incorporate other types of features depending on the applications.

video objects or backgrounds often exhibit nonuniform motions. Thus, applying to the next frame an appearance model constructed from the current one, will be problematic without correctly adapting it to the new position of the possibly deforming object/background caused by the motion.

Although various approaches have been proposed in recent years to tackle these problems, they either do not employ color models that are powerful enough to handle the appearance complexity, or do not adequately consider the motion complexity when updating the models across frames. We will analyze these limitations in more detail in the next section. As a result, the color models used in previous systems are often too rigid to handle video sequences with complex appearance and motion. Even with the help of other priors such as shape, pose, and structure, color is still an important feature in most natural videos, thus inaccurate color modeling often directly leads to segmentation errors. While these errors are correctable in an interactive setting, the user has to provide more manual input, which could be time consuming in many cases.

We introduce a new motion-adaptive color model called *Dynamic Color Flow*, or *DCF*. In this model, we combine motion estimation and color modeling into a single probabilistic framework that simultaneously addresses the appearance and motion complexities. The basic idea is to automatically and adaptively select the suitable color model, continuously ranging from a global model to a localized one, for different parts of the object, so that it can be reliably applied to segmenting future frames. The proposed framework does not assume accurate motion estimation. In fact, it takes into account the estimation errors and only assumes the motion estimation to be probabilistic, thus any motion algorithm with reasonable performance can be embedded into our system. Furthermore, we show how to apply the proposed DCF model to both foreground and background layers, leading to an efficient video cutout system as demonstrated by numerous examples.

## 1.1 Related Work

Video object segmentation is a classic problem that has been extensively studied for decades. Instead of surveying the large volume of literature, which is impractical here, we focus on classes of recent works that are most related to our system, and analyze their limitations, especially on color modeling.

**Global color models.** Some modern interactive video cutout systems use global color models, such as the popular choice of global Gaussian mixtures (GMM), to represent the appearance of the dynamic objects, e.g., [2–4]. Global color models do not consider the spatial arrangements of color components, thus are robust to object motion. However, the discrimination power of global models is too limited to deal with objects with complicated appearance.

**Pixel-wise color models.** The other extreme of color modeling is to consider every pixel on the image plane independently. Such method is often used in background subtraction systems. Assuming the camera is fixed and the background is static, these systems will form statistical models at every pixel location to describe the observed background colors, e.g., [5, 6]. However, using these models

require accurate frame-to-frame alignment, which may not be possible with dynamic background scenes.

**Localized color models.** The recently proposed SnapCut system [7] employs localized color models (see also [8]). It consists of a group of spatially constrained color components that are distributed along the object’s boundary in an overlapping fashion. Each color component includes a GMM with a fixed spatial domain. When propagated across frames, these local models are first pushed by optical flow vectors to arrive at new destinations, before being applied for local segmentation. It has been shown that by localizing the color models, the foreground object can be modeled more accurately, leading to efficient segmentations. Although in this approach motion estimation is used to move local color models across frames, it is treated independently from color modeling and classification. The scale (spatial domain) of all local color models are fixed without considering the underlying motion. This can cause two problems: when the local motion is strong (like a waving hand), optical flow may lose track, and the fixed window size will be too small to allow the localized color models to capture the object. On the other hand, for parts of the object where local motion is small, the window size may become too large to accurately model the foreground to background transition. We will demonstrate these problems with real examples in later sections.

**Bilayer segmentation.** Recently, significant success has been achieved for live speaker-background segmentation for video conferencing. Assuming a stationary background, the background cut system [9] uses a background contrast attenuation method to adaptively suppress the contrasts that belong to the background, making extracting the foreground easier. The *i2i* system, [10], avoids explicit motion estimation using a second order HMM model as a temporal (learned) prior on segmentation. These systems can efficiently segment a video in a constrained environment, but are hard to generalize for other types of videos, such as the examples shown in this paper.

## 2 Dynamic Color Flow

To explain the proposed *Dynamic Color Flow model (DCF)*, we first put aside the whole interactive video object segmentation workflow, and focus on the fundamental problem of *segmentation propagation*, that is, given a known correct foreground/background segmentation on frame  $t$ , how to use it to build accurate color models for segmenting the foreground/background on frame  $t + 1$ . Note that for now we do not distinguish between foreground and background, we will show in Section 3 how to apply the model to both regions.

Segmentation is trivial if an accurate motion vector field between frames is available: for every pixel on frame  $t + 1$ , we just trace it back to the previous frame and see whether it comes from the target region or not. However, a perfect motion vector field is almost impossible to compute in real world, and directly using it for segmentation will be erroneous. The DCF model proposed in our



system explicitly models the motion inaccuracy, and provides a probabilistic framework unifying the local colors of the object and their dynamic motion.

Let  $\Omega$  be the region of interest on frame  $t$  ( $\Omega$  can be foreground  $F$ , background  $B$ , or other object in case of multiple objects).  $\Omega$  contains  $|\Omega|$  pixels  $X_i$  ( $i = 1, 2, 3, \dots, |\Omega|$ ). Denote the position of pixel  $X_i$  as  $x_i$ . For each pixel  $X_i$  inside  $\Omega$ , we use the locally-averaged optical flow  $\mathbf{v}$  as the motion vector to predict its position in frame  $t + 1$ ,  $x'_i = x_i + \mathbf{v}$ <sup>3</sup>. Assuming the motion vector is not accurate enough, instead of using  $x'_i$  deterministically, we treat it as the center of a Gaussian distribution,

$$f_i(y) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{\|y - x'_i\|^2}{2\sigma_i^2}\right), \quad (1)$$

where  $y$  is a location in frame  $t + 1$ . The variance  $\sigma_i$  measures the fuzziness of the prediction. Its value is dynamically set for each pixel, as we will explain in the next section.

Let  $c_{X_i}$  be the color vector of pixel  $X_i$ . The probabilistic prediction propagates the colors in  $\Omega$  to the next frame and generates a distribution  $p(c, y|\Omega)$ , the probability of observing the color  $c$  at location  $y$  on frame  $t + 1$  given that all colors come from  $\Omega$  on frame  $t$ . The conditional color distribution at  $y$  is

$$p(c|y, \Omega) = \frac{p(c, y|\Omega)}{p(y|\Omega)}, \quad (2)$$

where  $p(y|\Omega) = \sum_{i=1}^{|\Omega|} p(X_i)p(y|x_i)$  is a spatial term independent of color, so we treat it as a normalization constant. Since  $p(c, y|\Omega)$  is contributed by all pixels in  $\Omega$ , it can be written as

$$p(c, y|\Omega) = \sum_{i=1}^{|\Omega|} p(X_i)p(c, y|X_i). \quad (3)$$

Since the predicted position of  $X_i$  is independent of its color,

$$p(c, y|X_i) = p(c|c_{X_i})p(y|x_i). \quad (4)$$

Then we have

$$p(c|y, \Omega) = \frac{\sum_{i=1}^{|\Omega|} p(X_i)p(c|c_{X_i})p(y|x_i)}{p(y|\Omega)}, \quad (5)$$

where  $p(c|c_{X_i})$  is the probability of observing color  $c$  on frame  $t + 1$  given the existence of  $c_{X_i}$  on frame  $t$ . Given the fact that colors of the same object may vary across frames due to illumination changes, compression, and noise, we model this as a 3-D Gaussian distribution with mean vector  $c_{X_i}$  and covariance matrix  $\Sigma$ , i.e.,  $p(c|c_{X_i}) = \mathcal{N}(c|c_{X_i}, \Sigma)$ . We will describe the explicit computation later.

<sup>3</sup> Similarly to [7], we average optical flow vectors locally to remove noise.

As previously defined,  $p(y|x_i) = f_i(y)$ . Assuming equal priors for every pixel,  $p(x_i) = 1/|\Omega|$ , then

$$p(c|y, \Omega) \propto \sum_{i=1}^{|\Omega|} f_i(y) \mathcal{N}(c|c_{X_i}, \Sigma). \quad (6)$$

From Eqn. (6) it is clear that  $p(c|y, \Omega)$  can be interpreted as a non-parametric density estimation of the color sample set  $\{c_{X_i} | i = 1, 2, \dots, |\Omega|\}$ . Each sample  $c_{X_i}$  is weighted by  $f_i(y)$ , which is the probability of  $c_{X_i}$  arriving at  $y$ . We observe that the color sample set encodes the motion estimation of the color samples across video frames, thus the model inherently fuses motion and appearance into a unified framework.

It is worth mentioning that there has been some formal studies on modeling the statistics of optical flow ([11], [12], [13], [14], [15]). Particularly, [12] studied its spatial properties and the brightness constancy error, resulting in a probabilistic model of optical flow. Our model is quite different in the following aspects. First, those works aim at improving optical flow estimation of natural images by considering learned prior distributions from ground truth training data, while our framework employs probabilistic methods on existing optical flow results for the purpose of generating more accurate color models for segmentation. Second, the learned statistics in [12] are global priors, while ours allows defining the distribution for individual pixels depending on the local motion. In fact, our model works with any optical flow algorithm that has reasonable performance and can certainly benefit from replacing the simple Gaussians by more accurate distributions as in [12].

Directly estimating  $p(c|y, \Omega)$  for each pixel location on frame  $t + 1$  is computationally expensive, therefore we employ the following approximations to efficiently speed up the computation. First, we use the  $Luv$  color space and assume class-conditional independence of the three channels,<sup>4</sup> thus  $p(c|y, \Omega)$  can be estimated as the product of three 1-D *PDFs* rather than a 3-D *PDF*, and the covariance matrix  $\Sigma$  in Eqn. (6) can be computed in each channel. Second, the 1-D *PDFs* at every  $y$  location are incrementally built using a quantized histogram containing 32 bins. Denoting the  $L$ -channel histogram at  $y$  as  $H_y^L$ , when propagating  $X_i$ , the  $L$  component of  $c_{X_i}$  with weight  $f_i(y)$ , is added to  $H_y^L$  for every  $y$  that is within a neighborhood centered at  $x'_i$  with a radius of  $R = 4\sigma_i$  (we then use a truncated Gaussian to replace the Gaussian function in Eqn. (1)).

After propagating all pixels within  $\Omega$ , we apply 1-D kernel density estimation, [16], on every histogram. Let now  $\bar{H}_y^L$  be the estimated density for the  $L$  channel at  $y$ ,  $u$  and  $v$  channels are similarly computed. Also, let us denote the color at  $y$  in frame  $t + 1$  as  $c_y = \{l, u, v\}$ . Finally, the probability of  $c_y$  coming from  $\Omega$  is

$$p(c_y|y, \Omega) = \bar{H}_y^L(l) \cdot \bar{H}_y^u(u) \cdot \bar{H}_y^v(v). \quad (7)$$

<sup>4</sup> Note that class-conditional independence is a weaker assumption than feature independence.

This procedure computes the probability for every pixel in the next frame  $t + 1$  once the parameters  $\sigma_i$  are given. Next we show that the model adaptively changes scales by using different  $\sigma_i$ -s.

**Global Color Model.** When all the  $\sigma_i \rightarrow \infty$ , all color samples are equally weighted, generating identical color distribution at each location, which is equivalent to a global color model. As mentioned earlier, this model only works well when the object has distinct colors from the rest of the scene, and is not affected by large motions which are hard to track.

**Localized Classifier.** Setting all  $\sigma_i$ -s to the same value  $r$ , we get a set of moving localized classifiers similar to the recently proposed SnapCut system [7]. This model assumes the object can be tracked reasonably well, i.e., the tracking error is less than  $4r$ .

**Stationary Pixel-wise Model.** When  $\sigma_i \approx 0$ , we have the pixel-wise color models commonly used in previous background subtraction systems ([5], [6], [9]). This model can be used if the video background is still or an accurate alignment can be achieved.

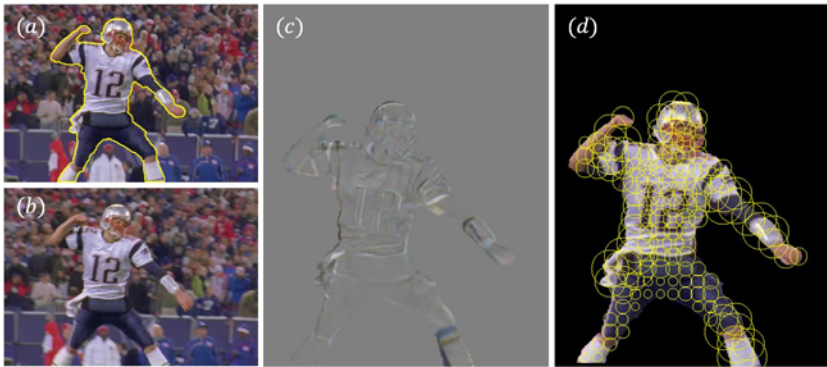
**Dynamic Model.** In all the above cases the motion scales of different parts of the object are assumed to be the same. However, we argue that most real world examples are likely to contain motions of mixed scales. For instance, for a walking person, his/her hand or foot motion obviously has a larger scale than his/her body. Thus, by dynamically determining  $\sigma_i$  for every pixel, our model offers the flexibility to adapt to different motion scales, even on the same object. This is a key advantage of the proposed DCF model. We will describe how to compute  $\sigma_i$  in the next section when we demonstrate how to apply this model to video segmentation.

### 3 DCF for Video Object Segmentation

In this section we apply the proposed DCF model to user-guided video object segmentation. We assume the video contains two independent foreground ( $F$ ) and background ( $B$ ) layers, although there is no fundamental limit on extending our model to multiple layers. The DCF model is applied to both  $F$  and  $B$  layers for a balanced modeling. The segmentation is then solved within a MRF framework.

#### 3.1 The Foreground Layer

The foreground object usually presents various local motion scales.  $\sigma_i$ , by its definition (see Eqn. (1)), is related to the prediction error of the foreground optical flow. For erratic movement where the optical flow is likely to contain large errors, we set  $\sigma_i$  to large values. For slow or stable motion, the optical flow is generally more reliable, thus the value of  $\sigma_i$  is reduced, yielding more localized color models which have greater classification power. In this way  $\sigma_i$  changes adaptively with the prediction error for the different parts of the object.



**Fig. 1.** (a) Frame  $t$  with known segmentation (*the yellow contour*). (b) Frame  $t + 1$ . (c) The difference image between the warped frame  $t$  and frame  $t + 1$ . (d) Values of  $\sigma_i$  (*yellow circles*), adapting to the local average intensity of (c) across the object.

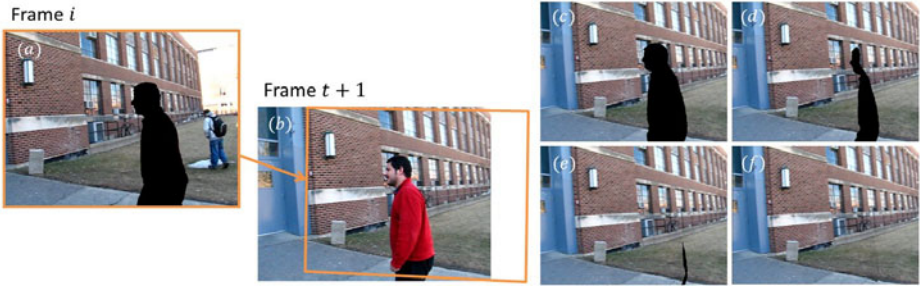
To compute the prediction error, the key frame is warped by the (locally averaged) optical flow to align with the next frame. In our system we define the alignment error  $e(x)$  as the local average of frames difference,  $e(x) = \sqrt{\frac{1}{m} \sum_{x \in N_x \cap \Omega'_F} \|I'_t(x) - I_{t+1}(x)\|^2}$ , where  $N_x$  is a square neighborhood centered at  $x$ ,  $I'_t$  and  $\Omega'_F$  are the warped color image and the binary foreground map from frame  $t$  to  $t + 1$  respectively, and  $m$  is the number of foreground pixels in  $N_x$ . Accurate alignment generally indicates reliable optical flow in the local regions, thus  $\sigma_i$  can be defined linearly proportional to  $e(x)$ . For flat, textureless regions where the local alignment error is always small, a lower bound term  $\sigma_{min}$  is added to increase robustness. Defining the local smoothness as  $s(x) = \frac{1}{1+\beta \cdot \bar{g}(x)}$ , where  $\bar{g}(x) = \sqrt{\frac{1}{m} \sum_{x \in N_x \cap \Omega'_F} |\nabla I_\sigma(x)|^2}$  is the local average of image gradient, and  $I_\sigma = I'_t * G_\sigma$ , we compute

$$\sigma_i = \begin{cases} \alpha \cdot e(x'_i) + s(x'_i) \cdot \sigma_{min}, & e(x_i) \leq e_{max}, \\ \alpha \cdot e_{max}, & e(x'_i) > e_{max}, \end{cases} \quad (8)$$

where  $\alpha \cdot e_{max}$  is the upper bound of  $\sigma_i$ . Typically  $\alpha = 0.2$ ,  $\beta = 10$ ,  $e_{max} = 50$ , and  $\sigma_{min} = 4$ . We will later show that this definition leads to improved results over traditional fixed  $\sigma_i$  color models (see Fig. 6), while our system is general to adopt more sophisticated estimation of  $\sigma_i$ . Compared to [7], where the colors are sampled within windows of a constant size, our algorithm uses a flexible sampling range that generates more accurate local color distributions. An example is shown in Fig. 1, where we can clearly see how  $\sigma_i$  changes based on local motion estimation errors.

### 3.2 The Background Layer

The background layer can be essentially treated in the same fashion as the foreground one. However, the occluded background behind the object is missing



**Fig. 2.** Consider that all the frames  $i$  prior to frame  $t + 1$  have been segmented. (a),(b) A frame with known background is warped to frame  $t + 1$  (working frame) using homography. (c)-(f) As additional prior frames are projected, the background is gradually completed, and (f) is used as the background layer for frame  $t + 1$ .

in frame  $t$ . In this section we explain two simple scenarios and methods to reconstruct the missing background. Note that our system is not limited to these two methods, and more complicated video mosaicking such as [17],[18], or hole filling algorithms such as [19], can be employed for more accurate background reconstruction.

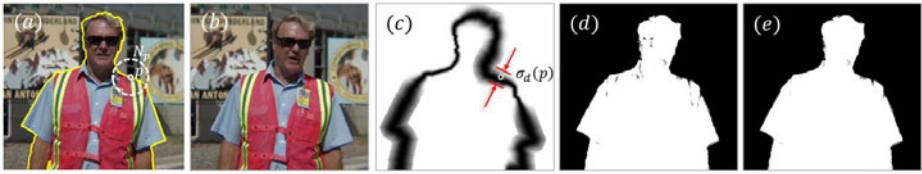
**A Clean Plate.** For videos which present a shot of the scene without the objects present, we can directly use the clean plate to build the background model. To deal with moving cameras, we estimate a homography by SIFT matching and RANSAC filtering, and then project the clean plate onto the current frame to be segmented. Similar to the foreground modeling, the DCF model is applied to the reconstructed clean plate, except that  $\sigma_i$  is fixed for every background pixel. Typically for static background  $\sigma_i$  is set to  $[2, 4]$  to compensate for small alignment errors.

**Progressive Background Completion.** In case a clean plate is not available, we use a progressive background completion method similar to the one proposed in [20]. Suppose the first  $t$  frames have been segmented, the segmented backgrounds are projected onto frame  $t + 1$  in a reverse order, from frame  $t$  to frame 1, recovering as much occluded background as possible. In general if the foreground object has a large relative motion against the background, a dynamic background plate can be quickly recovered as the segmentation process evolves. Such an example is shown in Fig. 2.

Once the DCF model is constructed for both the foreground and background layers, the foreground probability of a pixel  $y$  is computed as

$$p^C(y) = \frac{p(c_y|y, F)}{p(c_y|y, F) + p(c_y|y, B)}, \quad y \in I_{t+1}. \quad (9)$$

As demonstrated in [7], constructing an accurate probability map is the key to achieve accurate object segmentation. Compared with color models used in previous video segmentation systems, the DCF model produces more accurate results,



**Fig. 3.** The shape prior is a variable bandwidth border around the warped object contour (yellow curve). (a) For every point  $p$  on the contour, compute the average of histogram distance  $D(p)$  in the neighborhood  $N_p$ . (b) The next frame. (c) Shape prior function,  $p^S$  shown in gray scale from darkest (0) to brightest (1). Similar F/B color distributions result in narrow local bandwidth and tight shape constraint, and vice versa. (d) Foreground color probability  $p^C(y)$ . (e) Integrated shape and color foreground probability  $p(y)$ .

thanks to the motion adaptive local scale and improved background modeling. In Fig. 6 we will compare the color probability maps generated by DCF and those generated by simple background subtraction, global GMM color models, and the SnapCut system. We tested on difficult examples where foreground and background color distributions are highly overlapping, and the backgrounds are highly cluttered.

### 3.3 Segmentation with Shape Priors

Directly feeding the color probability map generated by Eqn. (9) (see also Fig. 6) to a graph cut optimization may still result in some small segmentation errors, since the color probability map tends to be noisy. To further improve the segmentation, we borrow the general idea from [7] of incorporating dynamic and local shape priors. The basic idea is to create a variable bandwidth contour adaptive to the local statistics of the DCF model. This is in spirit similar to the variable bandwidth trimap proposed in [4] for the purpose of image matting.

Let  $p$  be a point on the object contour (warped from the previous frame), and  $N_p$  a neighborhood centered at  $p$ . Define the distance between two histograms as  $d_H(\bar{H}_1, \bar{H}_2) \triangleq 1 - \sum_i \min\{\bar{H}_1(i), \bar{H}_2(i)\}$ . Let  $\bar{H}_{F,y}^L, \bar{H}_{F,y}^u, \bar{H}_{F,y}^v$  be the three foreground color histograms at a pixel  $y$ , and  $\bar{H}_{B,y}^L, \bar{H}_{B,y}^u, \bar{H}_{B,y}^v$  the corresponding background color histograms at  $y$ . Then, define

$$D(y) \triangleq \min\{d_H(\bar{H}_{F,y}^L, \bar{H}_{B,y}^L), d_H(\bar{H}_{F,y}^u, \bar{H}_{B,y}^u), d_H(\bar{H}_{F,y}^v, \bar{H}_{B,y}^v)\}, \quad (10)$$

and for added robustness, consider  $\bar{D}(p) \triangleq \frac{1}{K} \sum_{y \in N_p} D(y)$ , where  $K$  is the number of pixels in  $N_p$ . The local shape profile  $p^S(y) = 1 - \mathcal{N}(d_y | \sigma_d)$ .  $\mathcal{N}(d_y | \sigma_d)$  is then a Gaussian distribution with variance  $\sigma_d$ , which is linearly proportional to  $\bar{D}(p)$ , and with  $d_y$  the Euclidean distance from  $y$  to the contour point  $p$ . Larger  $\bar{D}(p)$  indicates that the local foreground and background colors are more separable, thus a wider shape profile is used to give less spatial constraint, and vice versa. Finally, the integrated probability at  $y$ , combining both local shape and color models, is defined as

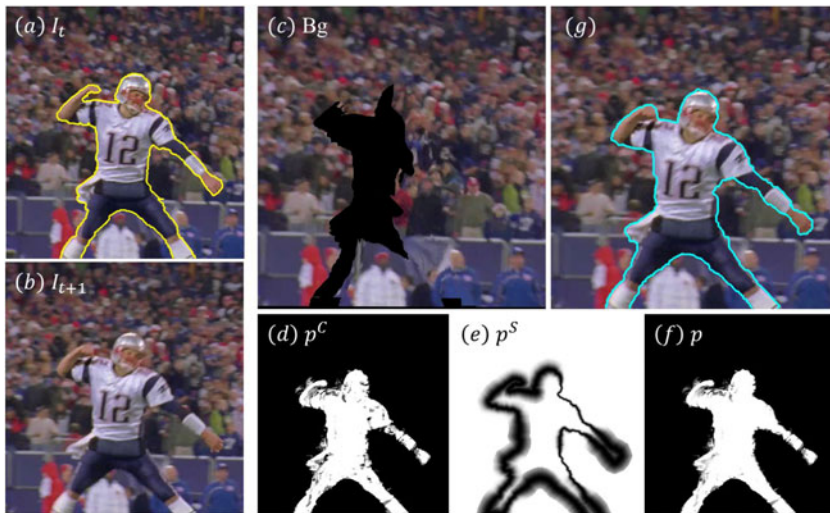
$$p(y) = p^C(y)(1 - p^S(y)) + M'_{t+1}(y)p^S(y), \quad (11)$$

where  $M'_{t+1}$  is the warped object mask with 1 inside and 0 outside. Essentially  $p^S(y)$  is used as a weight to linearly combine the color probability  $p^C(y)$  with the warped object mask  $M'_{t+1}$ . Please refer to [7] for more details of this equation. An example is shown in Fig. 3.

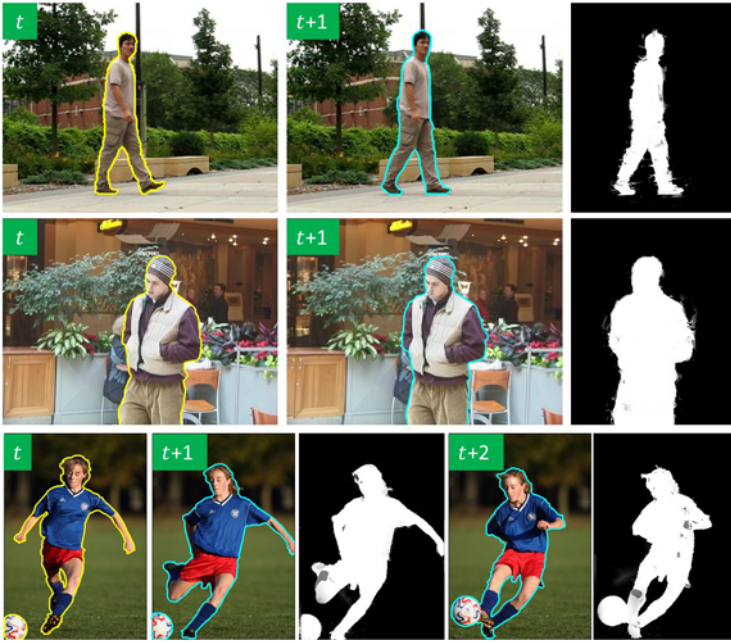
Using  $p(y)$  as the data term, and the image gradient statistics for the neighborhood term as proposed in [21], the current video frame  $t + 1$  is then segmented with a standard graph cuts image segmentation algorithm [22]. Examples are shown in figures 4 and 5. The user can optionally add scribbles to correct segmentation errors towards a more accurate segmentation, which then becomes the key frame for the next frame. This process is repeated until the whole sequence is segmented. Additionally, if necessary, the binary segmentation can be processed with a temporally-coherent matting algorithm, [7], producing soft alpha mattes for the foreground object for high-quality compositing tasks.

## 4 Experiments and Comparisons

We have tested our system on a variety of challenging video examples containing complex color distributions (figures 4 and 5(b)), highly cluttered background (figures 4 and 7), rapid topology changes (Fig. 5(c)), motion blur (Fig. 4), and camera motion (Fig. 2).



**Fig. 4.** Propagating the segmentation from frame  $t$  to  $t + 1$ . (a) Frame  $t$  with segmentation (yellow curve). (b) Frame  $t + 1$ . (c) The partially recovered background of frame  $t + 1$ . (d) Color probability  $p^C(y)$  shown in gray scale. (e) Shape prior derived from frame  $t$ . (f) Incorporating the shape prior further improves the quality of the probability map  $p(y)$ . (g) Final segmentation (cyan curve) of frame  $t + 1$  without any user interactions.



**Fig. 5.** Additional examples of segmentation propagation on objects with diverse motion. In each example we show the key frame  $t$  with segmentation (*yellow curves*), the computed segmentation on frame  $t + 1$  (*cyan curves*), and the probability maps in gray scale. In the last example, the segmentation propagates two consecutive frames.

First, Fig. 4 shows the intermediate results of segmenting one frame. Note the background reconstruction in (c) is only partially complete. For those pixels without background reconstruction colors, we simply sample nearby background colors for them in our current implementation, which already leads to satisfactory foreground estimation and segmentation, as shown in (d) and (g).

Then, Fig. 5 contains three additional examples that demonstrate different motion scales. In the first example, the walking person moves with dynamic (nonuniform) motion. The foreground in the second example is more stable but contains very complex colors (see supplementary material for the full video). The third example exhibits erratic motion and rapid topology changes that are very hard to track. Our system automatically adapts to these very different examples and produced accurate foreground probabilities that lead to high quality segmentation results.

We compared our proposed color model with background subtraction, global GMM, and the SnapCut system on two examples, as shown in Fig. 6. We used a basic background subtraction algorithm and manually selected the optimal threshold for each example. Due to the rigidity assumption for the static background and the lack of accurate foreground model, the algorithm is generally incapable of high quality segmentation tasks. The global GMM is without any





**Fig. 6.** Comparing color probability maps and segmentation results generated by simple background subtraction, the global GMM color model, the local color model from [7], and the proposed DCF on two examples. The gray scale images are the color probabilities generated by each method followed by their corresponding segmentation results. (For better visualization, images are cropped from original videos. See figures 3 and 7 for the full frames.)

doubt the least preferred in these examples, as both the foreground and background contain very similar colors. The SnapCut system improves the color probability results by localizing the color sampling. However, errors can occur if colors are confusing even in local regions, e.g., the black color in the glasses and in the background, first example. The DCF model generated more accurate color probabilities and segmentations for these examples.

To evaluate the complete interactive system, we compared our system with SnapCut on a video sequence, Fig. 7 (see supplementary material for additional sequences with comparisons in terms of segmentation accuracy and the amount of user interaction). Our system requires less user input to achieve comparable results. As the propagation progresses, the amount of interactions is further reduced thanks to the improved foreground and background models.

Of course our system cannot deal with all possible situations one may face in video segmentation. The DCF model assumes that all foreground colors on frame  $t + 1$  have been seen on frame  $t$ , thus cannot model newly appeared foreground colors due to occlusion and disocclusion, such as a self-rotating colored



**Fig. 7.** The football sequence, from left to right: frames 2, 5, 10, 13, 20. Frame 1 is pre-segmented. First row: segmentation (*red curves*) and user scribbles (*blue as foreground and green as background*) by the SnapCut system. Second row: segmentation (*yellow curves*) and user scribbles by our system. Third row: new composites on white.

ball where new colors constantly appear from one side of the object. The shape prior can only be used when the foreground shape is consistent and cannot be applied for things like fire and water. Also, if the background is highly dynamic, like a foreground person passing by a group of walking people, then the simple background construction methods described in Section 3.2 will fail. In these cases, more user input, or more advanced motion estimation and background reconstruction methods, will be needed to improve the performance of the system.

## 5 Concluding Remarks

A new color model that, unlike previous methods, incorporates motion estimation in a probabilistic fashion, was introduced in this paper. By automatically and adaptively changing model parameters based on the inferred local motion uncertainty, the proposed method accurately and reliably models the object appearance, and significantly improves the foreground color probability estimation. We applied the new model to both foreground and background layers for video object segmentation, obtaining significantly improved results when compared to previous state-of-the-art systems.

## References

1. Wang, J., Cohen, M.: Image and video matting: A survey. *Foundations and Trends in Computer Graphics and Vision* 3, 97–175 (2007)
2. Wang, J., Bhat, P., Colburn, A., Agrawala, M., Cohen, M.: Interactive video cutout. In: *Proc. of ACM SIGGRAPH* (2005)
3. Li, Y., Sun, J., Shum, H.: Video object cut and paste. In: *Proc. ACM SIGGRAPH*, pp. 595–600 (2005)
4. Bai, X., Sapiro, G.: A geodesic framework for fast interactive image and video segmentation and matting. In: *Proc. of IEEE ICCV* (2007)
5. Zivkovic, Z.: Improved adaptive Gaussian mixture model for background subtraction. In: *Proc. of ICPR* (2004)
6. Elgammal, A., 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)
7. Bai, X., Wang, J., Simons, D., Sapiro, G.: Video snapcut: robust video object cutout using localized classifiers. *ACM Trans. Graph.* 28, 1–11 (2009)
8. Price, B., Morse, B., Cohen, S.: Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In: *Proc. of ICCV* (2009)
9. Sun, J., Zhang, W., Tang, X., yeung Shum, H.: Background cut. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3952, pp. 628–641. Springer, Heidelberg (2006)
10. Criminisi, A., Cross, G., Blake, A., Kolmogorov, V.: Bilayer segmentation of live video. In: *Proc. of CVPR* (2006)
11. Roth, S., Black, M.J.: On the spatial statistics of optical flow. *IJCV* 74, 33–50 (2007)
12. Sun, D., Roth, S., Lewis, J.P., Black, M.J.: Learning optical flow. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008*, Part III. LNCS, vol. 5304, pp. 83–97. Springer, Heidelberg (2008)
13. Black, M.J., Yacoob, Y., Jepson, A.D., Fleet, D.J.: Learning parameterized models of image motion. In: *Proc. of CVPR*, pp. 561–567 (1997)
14. Simoncelli, E., Adelson, E.H., Heeger, D.J.: Probability distributions of optical flow. In: *Proc of CVPR*, pp. 310–315 (1991)
15. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/kanade meets horn/schunck: combining local and global optic flow methods. *IJCV* 61, 211–231 (2005)
16. Silverman, B.: *Density estimation for statistic and data analysis*. Monographs on Statistics and Applied Probability (1986)
17. Irani, M., Anandan, P., Bergen, J.: Efficient representations of video sequences and their applications. *Signal Processing: Image Communication* 8, 327–351 (1996)
18. Rav-Acha, A., Pritch, Y., Lischinski, D., Peleg, S.: Dynamosaicing: Mosaicing of dynamic scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*. 29, 1789–1801 (2007)
19. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: *Proc. of ACM SIGGRAPH*, pp. 417–424 (2000)
20. Chuang, Y.Y., Agarwala, A., Curless, B., Salesin, D., Szeliski, R.: Video matting of complex scenes. In: *Proc. of ACM SIGGRAPH*, pp. 243–248 (2002)
21. Rother, C., Kolmogorov, V., Blake, A.: Grabcut - interactive foreground extraction using iterated graph cut. In: *Proc. of ACM SIGGRAPH*, pp. 309–314 (2004)
22. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 23, 1222–1239 (2001)

# What Is the Chance of Happening: A New Way to Predict Where People Look

Yezhou Yang, Mingli Song\*, Na Li, Jiajun Bu, and Chun Chen

Zhejiang University, Hangzhou, China  
brooksong@zju.edu.cn

**Abstract.** Visual attention is an important issue in image and video analysis and keeps being an open problem in the computer vision field. Motivated by the famous Helmholtz principle, a new approach of visual attention analysis is proposed in this paper based on the low level feature statistics of natural images and the Bayesian framework. Firstly, two priors, i.e., Surrounding Feature Prior (*SFP*) and Single Feature Probability Distribution (*SFPD*) are learned and integrated by a Bayesian framework to compute the chance of happening (*CoH*) of each pixel in an image. Then another prior, i.e., Center Bias Prior (*CBP*), is learned and applied to the *CoH* to compute the saliency map of the image. The experimental results demonstrate that the proposed approach is both effective and efficient by providing more accurate and quick visual attention location. We make three major contributions in this paper: (1) A set of simple but powerful priors, *SFP*, *SFPD* and *CBP*, are presented in an intuitive way; (2) A computational model of *CoH* based on Bayesian framework is given to integrate *SFP* and *SFPD* together; (3) A computationally plausible way to obtain the saliency map of natural images based on *CoH* and *CBP*.

## 1 Introduction

The surrounding world contains a tremendous amount of visual information which the human visual system (*HVS*) cannot fully process [1]. Therefore, human tends to pay attention to only a few parts while neglect others in front of a scene. This phenomenon is usually called visual attention by psychologists. In order to predict where people look in an image automatically, visual attention analysis has been investigated for dozens of years in computer vision field. But till now it is still an open problem to be tackled. Recently, understanding computer vision problems from the view of psychologist is becoming an important track. As visual attention is also an important issue and has been studied for more than a century in the psychology field, it is reasonable to adopt some useful concepts of psychology to solve the visual attention analysis problem.

By treating the visual attention analysis as a signal processing problem, researchers believe *HVS* functions like a filter. It is natural for them to simulate the psychological mechanism by a filter computationally [2,3]. However, it is

---

\* Corresponding author.

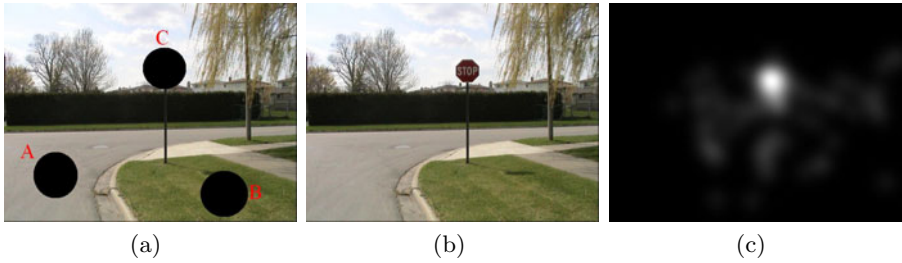
difficult to strictly simulate the visual attention mechanism in practice because the operating detail of *HVS* is still unknown until now even in the psychology field. Therefore, although some existing approaches [4,5,6] try to build up elegant mapping from psychological theories to computational implementation, they do not match the actual human saccade from the eye-tracking data and are computational consuming in practice.

Another way of approach tries to learn a visual attention model by taking into account of both the low level and the high level features based on the human eye tracking data. In [7], the pixel intensity is extracted as low level feature. And the semantic information is used as the high level features obtained by carrying out face detection, car detection, etc.. However, the study of *HVS* in [8] shows that the visual attention scheme is a kind of pre-processing before semantic analysis. Moreover, it's difficult to always extract the semantic information successfully from the given image.

Different from the aforementioned approaches, Bruce et al. [9,10] try to figure out the correlation between an information theory based definition of visual saliency and the fixation behavior of *HVS*. Then, in order to realize Bruce's theory in a computationally plausible way, some researchers [11,12,13] argue that some natural statistics of visual features, e.g., blue sky seldom exists in the lower part of a scene, should play an important role in the visual attention process. Some methods, such as Independent Component Analysis (ICA), are employed to extract the visual features to achieve the natural statistics. However, such visual features are complex and usually fail to estimate the visual attention efficiently. Moreover, given an image, *HVS* usually tends to pay attention to the central region of the image [14]. This center bias phenomenon of *HVS* is ignored in these methods, which often leads to mismatch between the experimental results and the eye tracking ground truth.

Inspired by the famous Helmholtz principle [15], a new concept called Chance of Happening (*CoH*) is introduced in this paper. As described literally, *CoH* of a point represents how likely this point will exist at a specific location of an image. And *CoH* is believed to be largely determined by a couple of priors, e.g., the relationship between the features of a specific point and its surroundings. These priors are learned from daily life experience by the *HVS*. In our approach, we try to learn these priors in a computationally plausible way and use them to estimate the *CoH* of a point. Both the *CoH* and the center bias are taken into account to compute the saliency map of the image.

The rest of the paper is organized as follows: section 2 introduces the motivation of the proposed approach. Section 3 describes two low level feature priors: Surrounding Feature Prior (*SFP*) and Single Feature Probability Distribution (*SFPD*). A Bayesian framework is introduced to compute the *CoH* from these two priors. In section 4, Center Bias Prior (*CBP*) is proposed and in section 5, a probabilistic framework is presented to compute the saliency map of an image by integrating the *CoH* and *CBP* together. Experiment is carried out in section 6. And we conclude in section 7.



**Fig. 1.** Example of relationship between Chance of Happening ( $CoH$ ) and visual attention. (a) Scene with hidden regions; (b) Original scene; (c) Saliency map computed directly from eye-tracking dataset.

## 2 Motivation

*“Whenever some large deviation from randomness occurs, a structure is perceived.”* [15]

–H. von Helmholtz (1821-1894), Psychologist, Germany

H. von Helmholtz, the famous Gestalt psychologist in Germany in 19th century, gave his famous description of human visual perception above, called Helmholtz principle. As a commonsense statement, the Helmholtz principle means that “we immediately perceive whatever could not happen by chance” [15]. This description inspires us with two cues: given an image, first, human’s visual attention depends heavily on the chance of happening ( $CoH$ ) which is estimated based on the previous experience of human visual system ( $HVS$ ); second, only the region which “could not happen by chance” tends to be “immediately perceived” by  $HVS$ , where “could not happen by chance” means the  $CoH$  of the region is small. Figure 1 demonstrates a typical instance of Helmholtz principle.

In Figure 1, for each blacked out region (left) labeled A, B or C,  $HVS$  estimates its  $CoH$  based on previous experience. For example, given the surrounding context of blue sky, the existence of dark red spot of region C is a large deviation from our expectation of color intensity distribution according to our previous experience. This large deviation leads region C to a small  $CoH$ . In contrast, the  $CoH$  of A and B are large because they preserve the consistency to their surroundings as predicted by  $HVS$  according to our previous experience. Recalling “we immediately perceive whatever could not happen by chance”, C will be “perceived immediately” as its  $CoH$  is small. The eye tracking experimental result (Figure 1(c)) also justifies this principle.

Besides  $CoH$ , researchers also find that  $HVS$  usually tends to pay attention to the center of an image [14,16]. This phenomenon is called center bias. In our approach, both  $CoH$  and the center bias are taken into account to carry out the visual attention analysis by computing the saliency map of an image.

Based on the discussion above, three priors, two for *CoH* and one for the center bias, are presented in our approach. And then they are integrated into a computational framework to perform the visual attention analysis.

### 3 From Low Level Feature Priors to *CoH*

As aforementioned, the visual attention scheme is a kind of pre-processing before the semantic analysis. And the semantic information is high-level which is difficult to extract robustly. In our approach, only the low level feature priors are taken into account to compute *CoH* towards the visual attention analysis.

#### 3.1 Single Feature Probability Distribution

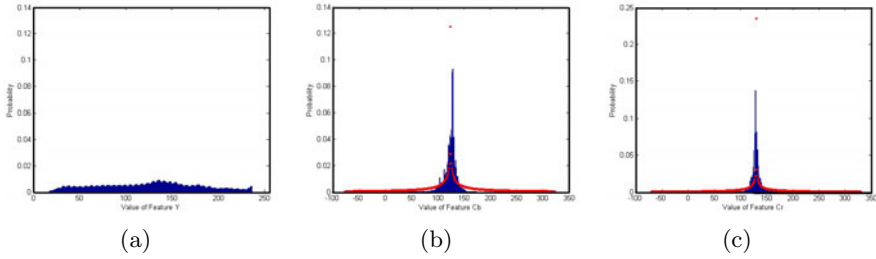
Color intensity is a kind of primary low level features used in computer vision society. In addition, color intensity can be manipulated more efficiently than other features. *YCbCr* is a preferable representation of the natural images in digital image coding system wherein *Y*, *Cb* and *Cr* are highly uncorrelated components corresponding to luminance, blue difference, and red difference [17]. So it is naturally to adopt *YCrCb* to learn the low level feature prior for the previous experience of *HVS* on color intensity distribution in natural images. Such low level feature prior is called Single Feature Probability Distribution (*SFPD*) in our approach.

In order to learn *SFPD*, we accumulate the occurrences of different intensities in the collected natural images for *Y*, *Cb* and *Cr* respectively. Thus the probability distributions of *Y*, *Cb* and *Cr* over the observed values is obtained correspondingly. Observing the curves in Fig. 2, we notice that the distribution of *Y* tends to be uniform, which means it doesn't take function in the *SFPD* computation. (Fig. 2(a)). In contrast to *Y*, the statistics of *Cb* (Fig. 2(b)) and *Cr* (Fig. 2(c)) look meaningful and Gaussian-like. Inspired by such observation, both the marginal distributions of *Cb* and *Cr* can be formulated as generalized Gaussian densities respectively:

$$P(Cb(x, y); \sigma_b, \theta_b) = \frac{\theta_b}{2\sigma_b\tau(\frac{1}{\theta_b})} \exp\left(-\left|\frac{Cb(x, y)}{\sigma_b}\right|^{\theta_b}\right) \quad (1)$$

$$P(Cr(x, y); \sigma_r, \theta_r) = \frac{\theta_r}{2\sigma_r\tau(\frac{1}{\theta_r})} \exp\left(-\left|\frac{Cr(x, y)}{\sigma_r}\right|^{\theta_r}\right) \quad (2)$$

where  $\tau$  is the gamma function,  $\sigma_{(\cdot)}$  is the scale parameter that describes the standard deviation of the density,  $\theta_{(\cdot)}$  is the shape parameter that is inversely proportional to the decreasing rate of the peak.  $Cb(x, y)$  and  $Cr(x, y)$  are the color intensity values of the pixel  $(x, y)$ . The model parameter  $(\theta_{(\cdot)}, \sigma_{(\cdot)})$  can be estimated using the moment matching method [18] or the maximum likelihood rule [19]. In our approach, the estimated parameters are:  $\sigma_b = 0.23$ ,  $\sigma_r = 0.041$ ,  $\theta_b = 0.26$ ,  $\theta_r = 0.22$ . The estimated distributions are depicted in Figure 2.



**Fig. 2.** (a)Single Feature Probability Distribution of Y (b)Single Feature Distribution of Cb (c)Single Feature Distribution of Cr. The blue bars represent the origin distribution and the red dots plot the estimated generalized Gaussian distributions

### 3.2 Surrounding Feature Prior

The surrounding context of a pixel is another important low level feature that reflects the contrast in an image. In our approach, the previous experience on the surrounding context in natural images is called Surrounding Feature Prior (*SFP*).

Given a surrounding window sized by  $w * w$ , two distances, i.e., the intensity distance and the location distance, are defined between pixel  $(x, y)$  and one of its surrounding pixels  $(x_j, y_j), j \in [1, w * w]$ . The definition of intensity distance in our approach is given below:

$$D_f^j(x, y) = |Y(x, y) - Y(x_j, y_j)| + |Cb(x, y) - Cb(x_j, y_j)| + |Cr(x, y) - Cr(x_j, y_j)| \tag{3}$$

where  $|\cdot|$  is the absolute operation.

And the location distance is defined as follows:

$$D_l^j(x, y) = \max(|x - x_j|, |y - y_j|) \tag{4}$$

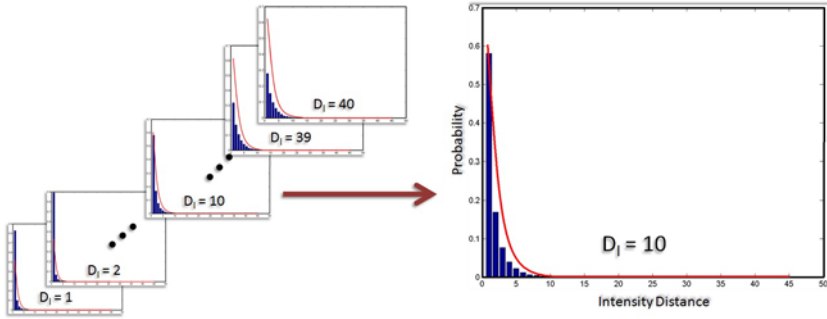
where  $\max(\cdot, \cdot)$  returns the largest value of the two inputs.

In order to learn the *SFP*, we count the number of pixels for each  $D_l$  varies from 1 to  $(w - 1)/2$  in the range of  $D_f$  based on the collected natural images. By observing Fig. 3, we notice that the probability distributions of intensity distance for different location distances are different from each other, but they are all exponential-like. So we use exponential function to fit these distributions using maximum likelihood rule. Given a pixel  $(x, y)$ , the existence probability of a pixel  $(x_j, y_j)$  in the surrounding window can be formulated as follows: :

$$P((x_j, y_j)|(x, y)) = \exp\left(-\psi\left(D_l^j(x, y)\right) D_f^j(x, y)\right) \tag{5}$$

where  $\psi(\cdot)$  is a enumerate function produces the unique coefficient for each location distance  $D_l^j(x, y)$  by exponential regression. In our approach, we set the surrounding window as  $81 * 81$ , and the estimated parameter  $\psi(\cdot)$  varies from 0.955 to 0.472 while the  $D_l$  increases from 1 to 40. The original and estimated distributions are depicted in Figure 3.





**Fig. 3.** Surrounding Feature Prior. The blue bars represent the origin distribution and the red lines are estimated exponential probability distribution.

### 3.3 Estimation of *CoH*

*CoH* of pixel  $(x, y)$  depends on not only the previous experience of color intensity distribution but also that of the surrounding context, i.e., *SFPD* and *SFP*. Thus *CoH* can be represented in a probability form and deduced based on Bayesian theorem as follows:

$$\begin{aligned}
 CoH(x, y) &= P(h(x, y)|\Omega(x, y)) \\
 &= P(\Omega(x, y), h(x, y))/P(\Omega(x, y))
 \end{aligned} \tag{6}$$

where  $h(x, y)$  in the first line means the happening of pixel  $(x, y)$ .  $\Omega(x, y)$  represents the pixel  $(x, y)$  and its surrounding window.

Suppose the  $w * w$  surrounding pixels are independent of each other. Since  $P(\Omega(x, y), h(x, y))$  is the joint probability representing the co-occurrence of the pixel  $(x, y)$  and its surrounding context, it can be computed as:

$$\begin{aligned}
 P(\Omega(x, y), h(x, y)) &= \prod_{j=1}^{w*w} (P((x_j, y_j), (x, y))) \\
 &= \prod_{j=1}^{w*w} (P((x_j, y_j)|(x, y)) P(x, y)) \\
 &= \prod_{j=1}^{w*w} (P((x_j, y_j)|(x, y)) P(Cb(x, y))P(Cr(x, y)))
 \end{aligned} \tag{7}$$

And  $P(\Omega(x, y))$  is the probability representing the occurrence of the surrounding context alone. So it can be computed as:

$$P(\Omega(x, y)) = \prod_{j=1}^{w*w} (P(x_j, y_j)) = \prod_{j=1}^{w*w} (P(Cb(x_j, y_j)) P(Cr(x_j, y_j))) \tag{8}$$

Substituting Eq. (7) and (8) into (6), we have

$$\begin{aligned}
 CoH(x, y) &= P(h(x, y) | \Omega(x, y)) \\
 &= \frac{\prod_{j=1}^{w*w} (P((x_j, y_j) | (x, y)) P(Cb(x, y)) P(Cr(x, y)))}{\prod_{j=1}^{w*w} (P(Cb(x_j, y_j)) P(Cr(x_j, y_j)))} \tag{9}
 \end{aligned}$$

### 4 Center Bias Prior

Besides the *CoH*, center bias is another important factor affects visual attention. Liu et al. [16] apply it by setting weight arbitrarily to compute the saliency map. However, it is still unclear that the decay rate of visual attention corresponding to the distance from the center. In order to obtain the decay rate away from the center, we learn a normal Bivariate Gaussian function from eye tracking dataset [10] to model the center bias in this paper. In the dataset, each image is accompanied with eye tracking data which are collected from 20 subjects free-viewing the image. Only 4 seconds are recorded during the free-viewing process by discarding the first several seconds which may introduce the imposed centering operation of the head-mounted eye tracking system. By accumulating all the fixation locations of human eyes in the eye tracking data, we depict the distribution of the fixation locations in Fig. 4. Observing the 3D depiction of the figure, a normal Bivariate Gaussian is defined to fit the eye fixation distribution as follows:

$$CBP(x, y) = \eta \exp \left( -\frac{1}{2(1-\rho^2)} \left( \frac{(x-\mu_1)^2}{\sigma_1^2} + 2\rho \frac{(x-\mu_1)(y-\mu_2)}{\sigma_1\sigma_2} + \frac{(y-\mu_2)^2}{\sigma_2^2} \right) \right) \tag{10}$$

where  $\eta = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}}$ . And we use  $\rho = 0$  for simplicity. Specifically, the image size used for *CBP* learning is  $511 \times 681$ . And the learned parameters of *CBP* are  $\mu_1 = 245.1, \mu_2 = 343.6, \sigma_1 = 91.2, \sigma_2 = 139.5$ .

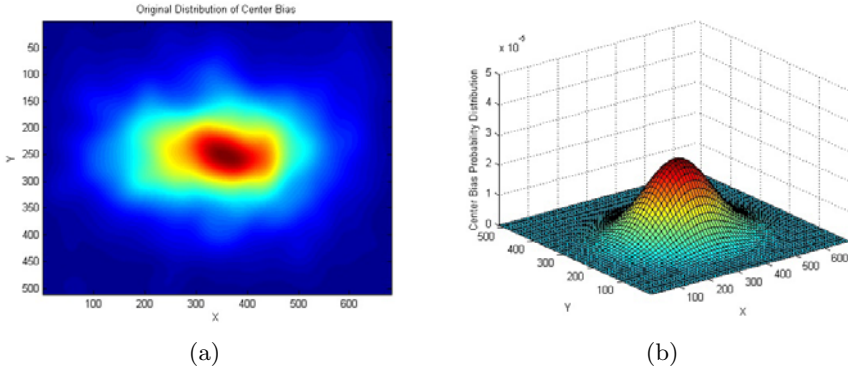
### 5 Computational Framework for Saliency

As aforementioned, the visual attention of human is influenced by not only the *CoH*, but also the center bias. So in our approach, both the *CoH* and the center bias are taken into account to compute the saliency map of the given image.

As discussed in Section 2, a point will be more attractive to the *HVS* when its *CoH* is relatively smaller. So in our approach, a pixel’s saliency is treated inversely proportion to *CoH*. Thus, the saliency  $S(x, y)$  of pixel  $(x, y)$  can be computed as follows:

$$S(x, y) = \lambda \frac{CBP(x, y)}{CoH(x, y)} \tag{11}$$

where  $\lambda$  is a scalar to perform adjustment.



**Fig. 4.** Center Bias Prior: (a)Origin Center Bias Distribution (b)Estimated Center Bias Distribution

By substituting eq. (9), (11) can be rewritten as:

$$S(x, y) = \lambda \frac{\prod_{j=1}^{w * w} (P(Cb(x_j, y_j))P(Cr(x_j, y_j))) \cdot CBP(x, y)}{\prod_{j=1}^{w * w} (P((x_j, y_j)|(x, y)) P(Cb(x, y))P(Cr(x, y)))}. \quad (12)$$

Since logarithm is a monotonically increasing function, we rewrite Eq. (12) to reduce the numerical computational complexity as follows:

$$\begin{aligned} S(x, y) &\rightarrow -\log\left(\frac{\prod_{j=1}^{w * w} (P((x_j, y_j)|(x, y)) P(Cb(x, y))P(Cr(x, y)))}{\prod_{j=1}^{w * w} (P(Cb(x_j, y_j))P(Cr(x_j, y_j)))}\right) + \log(CBP(x, y)) \\ &\rightarrow \left[ \begin{aligned} &\sum_{j=1}^{w * w} \left( \psi(D_l^j(x, y))D_f^j(x, y) - \left| \frac{Cb(x_j, y_j)}{\sigma_b} \right| \theta_b - \left| \frac{Cr(x_j, y_j)}{\sigma_r} \right| \theta_r \right) \\ &+ w * w * \left( \left| \frac{Cb(x, y)}{\sigma_b} \right| \theta_b + \left| \frac{Cr(x, y)}{\sigma_r} \right| \theta_r \right) \\ &- \frac{1}{2(1-\rho^2)} \left( \frac{(x-\mu_1)^2}{\sigma_1^2} + 2\rho \frac{(x-\mu_1)(y-\mu_2)}{\sigma_1\sigma_2} + \frac{(y-\mu_2)^2}{\sigma_2^2} \right) \end{aligned} \right] \quad (13) \end{aligned}$$

where “ $\rightarrow$ ” means “depend on”.

Fig. 5 depicts the flowchart of saliency map computation. For each input image, the RGB value of each pixel is transformed into YCbCr firstly. Then we calculate the saliency of each pixel by a computational framework in terms of Eq. (13). Thirdly, a  $25 \times 25$  Gaussian Blur is applied to remove the unwanted noisy saliency value. And histogram equalization is performed for better visualization.

## 6 Experiments

In our implementation, more than 1000 natural images are collected from internet so as to learn the two low level feature priors for *CoH*. And another prior CBP is learned based on one part of the eye tracking dataset [10] which consists

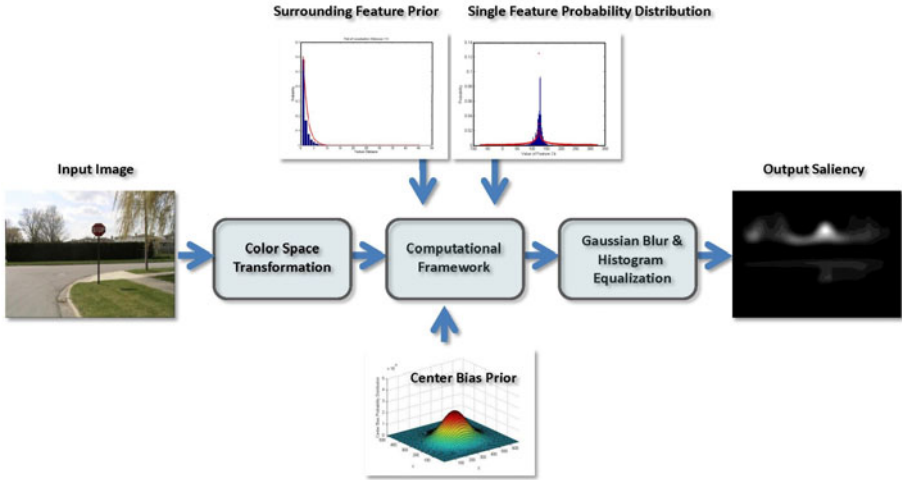


Fig. 5. The computational framework of our approach

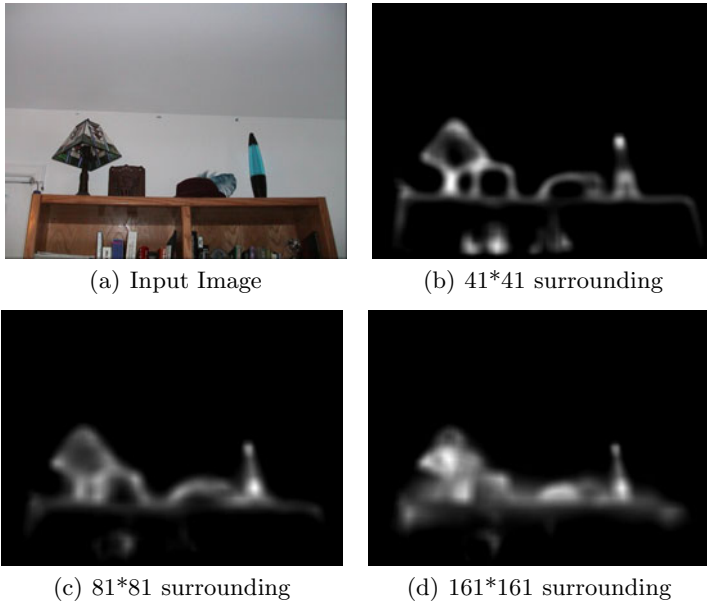
80 records. The other part of the eye tracking dataset is used as testing data for evaluation.

The computational complexity of our method is  $O(w^2 \times N)$  which is lower than most of the conventional approaches, e.g., Itti’s method.  $N$  is the number of pixels in the image. In addition, both look-up table and Integral Image [20] techniques are adopted by our approach to further speed up the computation of saliency map significantly. The experiment is carried out on a system with 2.8GHz processor and 8 Gigabyte memory. It takes our approach less than 1 minute on average for  $511 \times 681$  images with  $81 \times 81$  surrounding window. Under the same condition, it usually takes Itti’s method 10 minutes to deduce the saliency map.

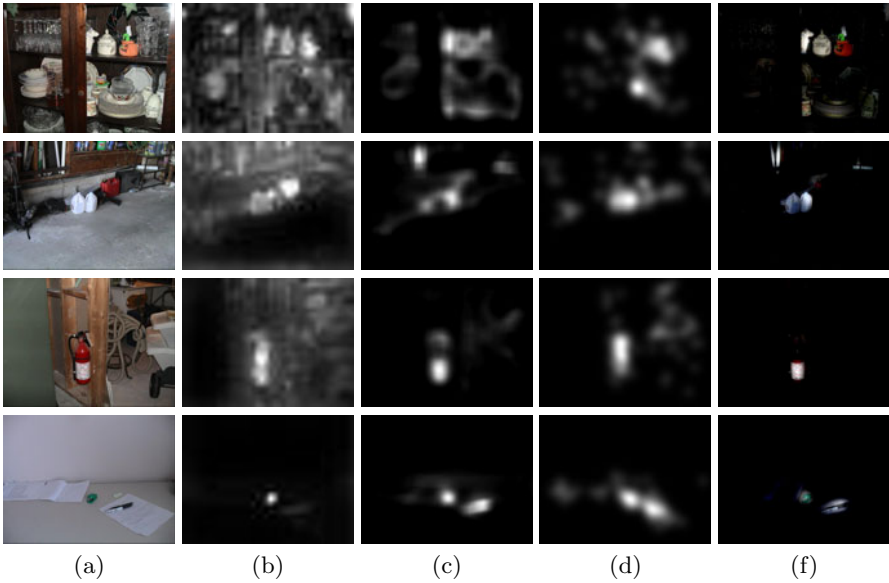
It is noticeable that different sizes of the surrounding window lead to different results (Fig. 6). When the surrounding window is too small, the computed saliency map usually stays away from the ground truth (Fig. 6(b)). On the contrary, the computed saliency map will be closer to the ground truth when the size of surrounding window becomes larger (Fig. 6(d)). But it will take more time to produce the result as a tradeoff. As a compromise, we choose  $81 \times 81$  for  $511 \times 681$  images in our approach (Fig. 6(c)).

### 6.1 Qualitative Evaluation

The saliency maps are depicted in Fig. 7 and 8 for the outdoor and the indoor images separately. The ground truths for evaluation are obtained from the eye tracking dataset. And the testing image are modulated by the output saliency of our method for better qualitative evaluation. It is noticeable that our method

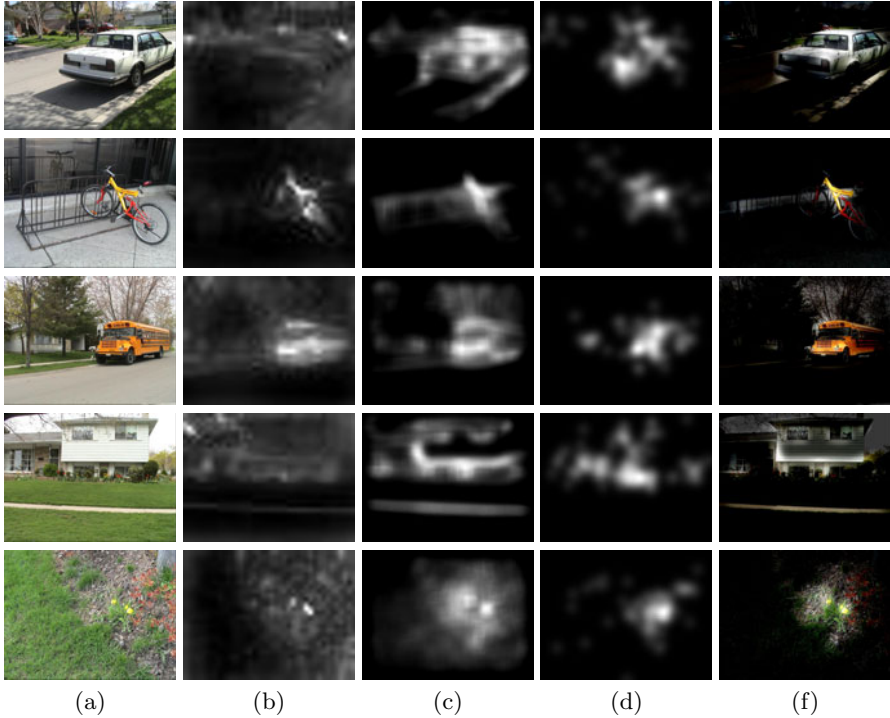


**Fig. 6.** Comparison of saliency maps computed by various surrounding sizes



**Fig. 7.** Saliency comparison on indoor images (a) Original image (b)Itti et al. saliency (c) Saliency of our method (d) Experimental saliency map (f) Modulated image

performs better than Itti’s method [4] on both the indoor and the outdoor images. Especially for the outdoor images, Itti’s method does not match the ground truth very well, e.g., the white car in the first row.

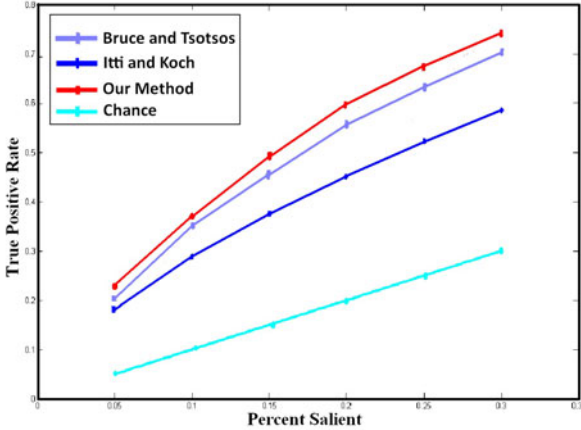


**Fig. 8.** Saliency comparison on outdoor images (a) Original image (b)Itti et al. saliency (c) Saliency of our method (d) Experimental saliency map (f) Modulated image

## 6.2 Quantitative Evaluation

Recently, the receiver operating characteristic curve (*ROC*) [7,10] becomes a widely adopted metric to evaluate eye fixation prediction quantitatively. The *ROC* metric treats the saliency map as a binary classifier in the image, wherein pixels with saliency value larger than a threshold are identified as “fixated” while the others “non-fixated”. By varying the threshold and testing the true positive rate for each threshold, an *ROC* curve can be drawn and the area under the curve indicates how well the saliency map predicts eye’s fixations.

Figure 9 shows the comparison between the proposed method and the conventional ones. In our evaluation, the threshold of visual attention region varies from top 5% to top 30% part of the saliency map. The *ROC* curves in the figure depict quantitatively that the proposed method takes advantage over the conventional ones.



**Fig. 9.** The ROC curves of performances for Itti [4], Bruce/Tsotsos [13] and our method. We also plot chance for comparison.

## 7 Conclusions

In this paper, a new computational approach is presented for visual attention analysis. Two low level feature priors, *SFPD* and *SFP*, are learned based on natural images and integrated by a Bayesian framework to compute the *CoH* at each pixel. Then another prior, *CBP*, is learned from the eye tracking dataset. Finally, the saliency of each pixel is obtained by taking consideration of both *CoH* and *CBP*. By using the proposed approach, visual attention analysis becomes more effective and more efficient than the existing low level feature based approaches and produces better matching to human eye-tracking data.

We also notice some limitations of our method. For example, the size of the surrounding window is set arbitrarily. In the future, we will develop a multi-scale approach to figure out the optimal size of the surrounding window. In addition, more low level feature priors for *CoH*, especially those biologically plausible priors, will be investigated. It may also be possible to employ a GPU-based implementation to parallelize the computation. However, these points do not impact on the conclusions of this paper or the theory presented.

## Acknowledgment

This paper is supported by the National Natural Science Foundation of China under Grant 60873124, by the Natural Science Foundation of Zhejiang Province under Grant Y1090516, and by the Fundamental Research Funds for the Central Universities under Grant 2009QNA5015.

## References

1. Tsotsos, J.: Analyzing vision at the complexity level. *Behavioral and Brain Sciences* 13, 423–445 (1990)
2. Wolfe, J.M., Cave, K.: *Deploying visual attention: The guided search model*. John Wiley and Sons Ltd., Chichester (1990)
3. Tsotsos, J.K., Culhane, S.M., Wai, W.: Modeling visual attention via selective tuning. *Artificial Intelligence* 78, 507–545 (1995)
4. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence* (1998)
5. Itti, L., Koch, C.: A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research* 40, 1489–1506 (2000)
6. Yang, Y., Song, M., Li, N., Bu, J., Chen, C.: Visual attention analysis by pseudo gravitational field. *ACM Multimedia*, 553–556 (2009)
7. Judd, T., Ehinger, K., Durand, F., Torralba, A.: Learning to predict where humans look. In: *International Conference on Computer Vision*, Acceptance (2009)
8. Koch, C., Ullman, S.: Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurobiology* 4, 219–227 (1985)
9. Bruce, N.D.B.: Features that draw visual attention: An information theoretic perspective. *Neurocomputing* 65–66, 125–133 (2005)
10. Bruce, N.D.B., Tsotsos, J.K.: Saliency based on information maximization. *Advances in Neural Information Processing Systems* 18, 155–162 (2006)
11. Itti, L., Baldi, P.: Bayesian surprise attracts human attention. *Advances in neural information processing systems* 19, 547–554 (2006)
12. Zhang, L., Tong, M.H., Marks, T.K., Shan, H., Cottrell, G.W.: Sun: A bayesian framework for saliency using natural statistics. *Journal of Vision* 8, 1–20 (2008)
13. Bruce, N.D.B., Tsotsos, J.K.: Saliency, attention, and visual search: An information theoretic approach. *Journal of Vision* 9(3), 1–24
14. Meur, O.L., Callet, P.L., Barba, D., Thoreau, D.: A coherent computational approach to model bottom-up visual attention. *IEEE Trans on Pattern Analysis and Machine Intelligence* 28, 802–816 (2006)
15. Desolneux, A., Moisan, L., Morel, J.: *From Gestalt Theory to Image Analysis, A Probabilistic Approach*. Springer Science and Business Media, Heidelberg (2008)
16. Liu, H., Jiang, S., Huang, Q., Xu, C., Gao, W.: Region-based visual attention analysis with its application in image browsing on small displays. *ACM Multimedia* (2007)
17. Zhai, G., Chen, Q., Yang, X., Zhang, W.: Scalable visual sensitivity profile estimation. In: *ICASSP*, pp. 876–879 (2008)
18. de Wouwer, G.V., Scheunders, P., Dyck, D.V.: Statistical texture characterization from discrete wavelet representations. *IEE Trans. Image Processing* 8, 592–598 (1999)
19. Poor, H.V.: *An Introduction to Signal Estimation and Detection*, 2nd edn. Springer, Heidelberg (1994)
20. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 511–518 (2001)



# Supervised and Unsupervised Clustering with Probabilistic Shift

Sanketh Shetty and Narendra Ahuja

Department of Electrical and Computer Engineering,  
University of Illinois, Urbana-Champaign, Urbana, IL 61801 USA  
{sshetty2,n-ahuja}@illinois.edu

**Abstract.** We present a novel scale adaptive, nonparametric approach to clustering point patterns. Clusters are detected by moving all points to their cluster cores using shift vectors. First, we propose a novel scale selection criterion based on local density isotropy which determines the neighborhoods over which the shift vectors are computed. We then construct a directed graph induced by these shift vectors. Clustering is obtained by simulating random walks on this digraph. We also examine the spectral properties of a similarity matrix obtained from the directed graph to obtain a K-way partitioning of the data. Additionally, we use the eigenvector alignment algorithm of [1] to automatically determine the number of clusters in the dataset. We also compare our approach with supervised [2] and completely unsupervised spectral clustering [1], normalized cuts [3], K-Means, and adaptive bandwidth meanshift [4] on MNIST digits, USPS digits and UCI machine learning data.

**Keywords:** Data Clustering, Image Segmentation.

## 1 Introduction

This paper is about automatic clustering with minimal user input. A cluster is viewed as a set of contiguous points having similar local point structures, defined by the point density, which are in contrast with their immediate surround. We allow clusters defined by a variety of global density-based criteria. (1) A cluster may consist of uniformly distributed points (having constant point density), or it may be characterized by a uniform density gradient, or it may be uniform in higher order derivatives of the density. (2) The gradient may be uniform along an open curve, giving rise to a uniform cluster. Alternately, an iso-density curve may be a closed contour in which case the cluster is modal, with a point of density extremum, surrounded by a succession of iso-contours with monotonically changing density. (3) A cluster may be of the same dimensionality as the underlying point pattern, or it may be confined to a subspace. (4) The defining criteria from (1) above, and other properties such as sizes, shapes and densities are unknown.

The basic idea of the proposed approach is to identify overlapping neighborhoods of points across the pattern, each completely contained within a cluster. Regardless of cluster type, we characterize these neighborhoods as density

isotropic. Clearly, when a cluster has gradually varying density, the neighborhood size will be smaller - small enough to pass as isotropic within the tolerance level being used to test for density isotropy. Thus a cluster with arbitrarily complex, smoothly varying spatial density will be composed of overlapping neighborhoods each of whose size will be inversely proportional to the rate of local density change. Clustering then amounts to finding and distinctly labeling each connected set of overlapping, uniform-density neighborhoods. The connected components are extracted by letting each cluster implode to a dense core in its interior, thus resulting in as many well separated and uniquely labeled cores as the number of clusters. This is done by gradually moving each point within each cluster towards its core, by identifying a shift vector associated with the point which is directed towards the cluster core.

## 2 Related Work

Clustering algorithms are extremely diverse in their definition of clusters and approaches to finding them. Recent surveys of clustering algorithms are present in papers by Jain et al. [5] and Xu and Wunsch [6]. We restrict our discussion to algorithms relevant to motivating our approach, in particular the X-shift family of algorithms and spectral clustering algorithms.

The works of Fukunaga and Hostetler [7] and Koontz et al. [8] are early examples of clustering algorithms based on computing local density gradients. These techniques were rediscovered by the computer vision community in the recent past and applied to a host of problems in clustering image and video data. More recently methods based on computing a shift-vector based on mean [4], medoid [9] or median [10] of a point neighborhood have been proposed. The key idea is to compute a point or exemplar along the density gradient to which a point is shifted. Their advantages are that they are unrestricted in the shapes of the clusters and also automatically determine the number of clusters. The medoid-shift algorithm can also be applied to cases where only the distances or similarities between data is available. However, only adaptive bandwidth meanshift [11] addresses the problem of scale selection. Adaptive scales at individual points are computed using a pilot kernel density estimate obtained at a fixed scale  $K$ . We found the final clustering to be sensitive to this value of the initial bandwidth (see sec 5). Additionally, heuristics for merging modes and minimum cluster size significantly affect the final clustering. We differ from the X-shift algorithms in how our shift vector is computed. The X-shift algorithms move points along the density gradient towards the mode. However, they are not sensitive to other types of local density disparities that may exist in the data, e.g. a density step. This is because they rely on decisions that are local to a point neighborhood. In contrast, we rely on evidence accumulation from relevant adjacent neighbors to decide the local shift. X-shift methods are also likely to fail for clusters with uniform point distribution as a unique density mode is unlikely to exist. They return an oversegmented result for such clusters. In contrast, we model the isotropy of point distributions in local neighborhoods. We propose a statistical

testing approach to detect density isotropy. We are sensitive to relevant density changes e.g. cluster boundaries, density steps, and density gradients while ignoring incidental density disparities that may arise due to sampling, e.g. points in a uniform cluster. Subsequently, we use these detected density isotropic regions to determine a valid neighborhood over which each point influences its neighbors to shift in its direction. In section 5, we show qualitative and quantitative experiments that compare our shift vectors against those of X-shift algorithms.

Spectral approaches [12,23,1], involve the analysis of the graph Laplacian to obtain an embedding using its eigenvectors. Following this, regular K-means clustering or thresholding is applied to the embedded points to obtain a final clustering. Ng et al. [2] propose analyzing the symmetric, normalized graph Laplacian to obtain an embedding. The normalized cuts algorithm, in contrast can be viewed as analyzing eigenvectors of the transition probability matrix of a random walk on the undirected graph induced by the points [13]. Zelnik and Perona [1], address the problems of scale selection and automatic determination of the number of clusters for spectral clustering. The key advantage of these approaches lies in their ability to model clusters of unrestricted shapes in any subspace of the original space. However, Nader and Galun [14] construct several failure cases of such approaches, including the self-tuning spectral clustering algorithm. In particular, they identify problems with the scale selection parameter when there is a significant difference in density between adjacent clusters of different sizes. Additionally, these algorithms are sensitive to outliers in the dataset. We use the shift vectors computed using our approach to define a probabilistic directed graph. We analyze the spectral properties of affinity matrices derived from this digraph to obtain our final K-way and unsupervised clustering. This may be viewed as spectral clustering using an alternative graph construction technique. We demonstrate that this alternate construction, utilizing properties of shift vectors rather than K-nearest neighbors similarities, outperforms spectral clustering algorithms on real datasets.

### 3 Approach

Our proposed approach is an extension of the concept of the force transform, introduced in [15] for image analysis, to point sets in  $\mathbb{R}^N$ . The force transform produces a vector at each pixel, which represents the direction and magnitude of attraction experienced by the pixel from the rest of the image [15]. Region borders are identified as adjacent points with divergent vectors, whereas region skeletons are identified as adjacent points with convergent vectors. These vectors are computed at a set of spatial and image intensity scales, which are then used to produce a hierarchical image segmentation. Here our goal is to label points belonging to the cluster interior and border analogous to pixel labeling in image regions.

There are two major parts to our approach. The first part has to do with the detection of isotropic density neighborhoods. To this end, we use a test to determine if the neighborhood has isotropic point distribution in it. The

second part has to do with labeling connected components formed by overlapping isotropic density neighborhoods. This is made more complex than it may appear by the possibility of false detection or false rejection of an isotropic neighborhood, which may lead to cluster splits, e.g., in the neck area of a cluster, or cluster merges, e.g., in locally isotropic appearing neighborhoods between two distinct clusters. Although, postprocessing could be performed to detect and correct such errors, we have developed a formulation which avoids the need for such postprocessing by posing the problem as one of robust signal detection amidst noise in the first place. The signal here is the connected neighborhoods and the noise is deviations from density isotropy. We achieve this by iteratively, gradually and probabilistically shifting each point towards its cluster interior. This itself is done in two steps: by computing the local direction for shift, i.e., towards cluster interior, and then identifying the cluster (core) from these shift vectors.

Consequently, there are three major steps in our approach: (1) detection of density isotropic neighborhoods, (2) computation of shift vectors, and (3) identification of clusters utilizing probabilistic shift. The following subsections describe how we formulate each of these steps.

### 3.1 Detection of Isotropic Density Neighborhoods

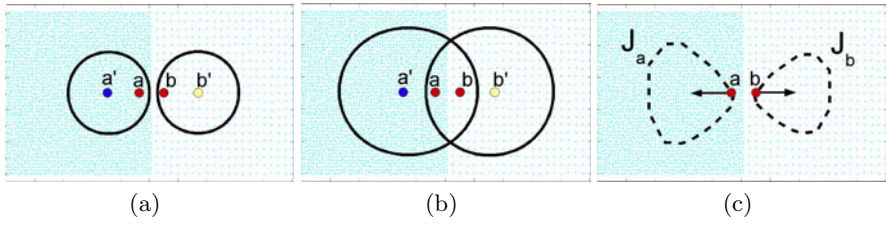
Our motivation for relying on isotropic density neighbors as the fundamental structures for clustering is as follows. It is reasonable to associate points within an isotropic density neighborhood with the same cluster. In contrast, density anisotropy, usually associated with a cluster boundary, indicates a plausible change in the cluster labels within a neighborhood. Therefore, density isotropy by itself may be used as a criterion for grouping points into clusters. However, we demonstrate that it is more useful as a scale selection criterion for computing shift vectors.

**Force Criterion:** We model the expected behavior of the force criterion [15] in isotropic and anisotropic neighborhoods to design a statistical testing approach to detect them. Figure 1(a) shows examples of isotropic density neighborhoods of a point. Figure 1(b) shows examples of anisotropic density neighborhoods of a point.

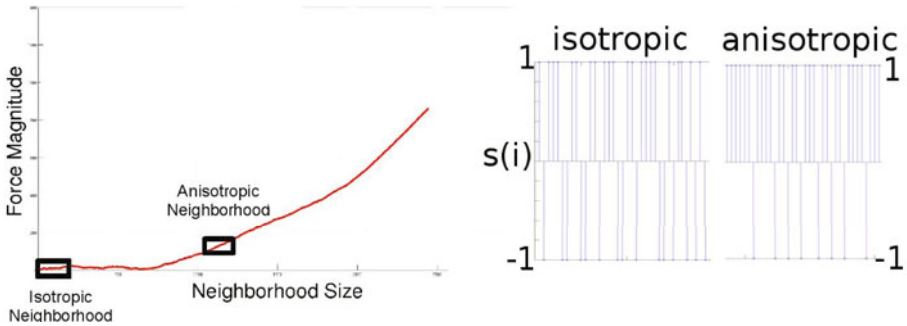
Given a set of points  $\{\mathbf{x}_i\}_{i=1}^n$ , centered at a point  $\mathbf{y}$ , and a weighting function  $w(\|\mathbf{y} - \mathbf{x}\|)$ , the force vector at  $\mathbf{y}$  is computed as:

$$\mathbf{f}_n(\mathbf{y}) = \sum_{i=1}^n w(\|\mathbf{y} - \mathbf{x}_i\|) * \frac{(\mathbf{x}_i - \mathbf{y})}{\|\mathbf{x}_i - \mathbf{y}\|} \quad (1)$$

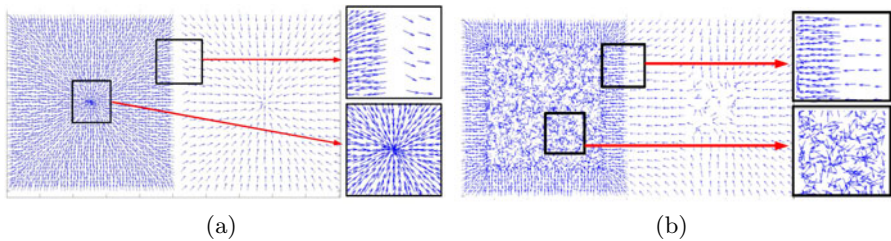
There are several possible choices for the weight function,  $w(\|\cdot\|)$ . The only requirement is that it is non-increasing [15]. We denote the magnitude of the force over the  $n$ -nearest neighbors as  $f_n(y) = \|\mathbf{f}_n(\mathbf{y})\|$ . Therefore, the set  $\{f_i\}_{i=1}^K$  represents the magnitude of the force vector computed over increasing neighborhood sizes. We use this set to develop our criterion for detecting isotropic neighborhoods. A non-zero magnitude for the force vector indicates anisotropy



**Fig. 1.** Given two clusters with a density step between them, we show (a) isotropic density regions for points  $a'$  and  $b'$ , to which points  $a$  and  $b$  belong. (b) However, the region for  $a'$  containing  $b$  has anisotropic density. Therefore,  $b$  does not belong to the influence neighborhood of  $a'$ . Similar reasoning holds for  $a$  and  $b'$ . (c) We show the sets  $J_a$  and  $J_b$  that contain  $a$  and  $b$  respectively in their influence neighborhoods. The shift is computed as a vector sum of influences of points in  $J$ .



**Fig. 2.** (left) Plot of the force criterion from equation  $\square$  over increasing neighborhood sizes. (right) Plot of random variable  $s_i$  for an isotropic density region and anisotropic density region.



**Fig. 3.** (a) Shift vectors computed for the entire point set. Notice the shift vectors diverging at the cluster boundary and converging at the center. (b) Mean shift vectors for the same point set. They do not respect the density step between clusters and are arbitrarily oriented in the cluster interior. This results in cluster fragmentation.

in the distribution of points in the neighborhood. If the magnitude continues to increase as we grow the neighborhood around the point, it symptomatic of a growing anisotropy in the local point distribution. The force vector points in the direction of increasing point density. However, if the point distribution is symmetric we expect the magnitude to fluctuate. Figure 2 shows a plot of the force vector for different neighborhood sizes around a point of interest.

We define a random variable  $s_i = \text{sign}(f_i - f_{i-1})$ . This represents the sign of the difference of force magnitudes computed at two adjacent neighborhood sizes. We claim that in a region with isotropic point density the distribution of  $s_i$  is uniform at its two possible values  $\{-1, 1\}$ . In an isotropic region the force magnitude is as likely to increase as it is to decrease. Any anisotropy in the neighborhood is incidental unless it is statistically significant. Formally, we propose the identification of isotropic neighborhoods as a detection problem.

$$H_0 : \{s_i\}_{i=1}^K \text{ has zero median} \tag{2}$$

$$H_1 : H_0 \text{ is false} \tag{3}$$

We test for  $H_0$  and  $H_1$  using the sign test that this distribution has a zero median [16]. If  $H_0$  is true it indicates an isotropic distribution of points in the given  $K$ -neighborhood. This test is performed at a significance level  $\alpha$ . Therefore, for increasing neighborhood sizes we perform the sign test on the computed force magnitudes and return the first point of failure as the neighborhood size,  $K_i$ , over which the current point has influence. This is defined as the *influence neighborhood* of a point and is used to compute the shift vectors at points contained in it.

### 3.2 Shift Vector Computation

Let  $J_i$  denote the set of indices of points for which  $\mathbf{x}_i$  appears within their respective neighborhoods of influence. It is reasonable for each point in  $J_i$  to assume that  $\mathbf{x}_i$  shares its cluster label. However, it is also possible that for some  $\mathbf{x}_i$ ,  $J_i$  has points from adjacent clusters, e.g., consider the case of points at the cluster boundary between two overlapping Gaussians. Therefore, we develop an approach where points in  $J_i$  compete for ownership of  $\mathbf{x}_i$ . The shift vector is the outcome of this competition. Given  $J_i$  the shift vector at a point is computed as:

$$\mathbf{a}_i = \sum_{j \in J_i} w(\|\mathbf{x}_i - \mathbf{x}_j\|) * \frac{(\mathbf{x}_j - \mathbf{x}_i)}{\|\mathbf{x}_j - \mathbf{x}_i\|} \tag{4}$$

Here  $w(\|\cdot\|)$  is some non-increasing weighting function. In our experiments we used the triangular weighting function ( $w\|\cdot\| \propto 1 - \frac{\text{dist}_i}{\max_{j \in J_i} \text{dist}_j}$ , if  $j \in J_i$ , else 0). It is important to recognize the difference between the force vector  $\mathbf{f}$ , in section 3.1, and the shift vector  $\mathbf{a}$ . The force vector, similar to X-shift vectors, points in the direction of the density gradient in the local neighborhood, as it is a purely local measure. In contrast, our shift vector,  $\mathbf{a}$ , points in the direction of greatest

agreement with local neighborhood properties. It points in the direction of the cluster whose points find the current point in most agreement with their local point distributions. This is important in our model of clustering as we seek to integrate neighborhoods with similar density properties while being sensitive to density discontinuities. Figures 3(a) and 3(b) further emphasize the advantages of our approach.

### 3.3 Cluster Identification

Cluster identification by connected components labeling of overlapping uniform neighborhoods has been proposed in [17]. However, as stated at the beginning of this section, this may lead to cluster splits and merges. Our shift vectors allow for a more informed connected components labeling. Shift vector at a point is directed in the general direction of the cluster core. We propagate labels in the general direction of the shift vector. We construct a probabilistic directed graph by connecting each point to other points in its influence neighborhood that lie in the half-space in the direction of its shift vector. Points are shifted probabilistically along this graph to cluster cores where the final clusters are obtained. This is realized using the interpoint transition probability matrix for the points defining the digraph.

**Constructing the Transition Probability Matrix.** First, define a directed graph  $G = \langle X, P \rangle$ , composed of a node set  $X = \{\mathbf{x}_i\}_{i=1}^n$  and a transition probability matrix  $P = \{p_{ij}: \text{probability of a transition from node } i \text{ to node } j\}$ . Given a node  $\mathbf{x}_i$ , its shift vector  $\mathbf{a}_i$ , and its influence neighborhood set  $K_i$ , we define a variable  $t_{ij} \in [0, 1]$  which represents the preference for moving from node  $i$  to node  $j$ . Formally, it is defined as:

$$t_{ij} = \max(0, w(|\mathbf{x}_j - \mathbf{x}_i|) \langle \mathbf{a}_i, \mathbf{x}_j - \mathbf{x}_i \rangle) \tag{5}$$

This produces positive values for nodes in the positive half space of the hyperplane defined by  $\mathbf{a}_i$  at  $\mathbf{x}_i$  and 0 otherwise. From this we obtain the probability of transitioning to a node  $\mathbf{x}_j, j \in K_i$ :

$$p_{ij} = \frac{t_{ij}}{\sum_{j \in K_i} t_{ij}} \tag{6}$$

Consider a point  $\mathbf{x}_b$  near the cluster border. Its shift vector, by construction, points towards the cluster interior. Therefore, for a point  $\mathbf{x}_j$ , within its influence neighborhood in the cluster interior, there is a non-zero probability of a transition from  $\mathbf{x}_b$  to  $\mathbf{x}_j$ . However, the reverse is not necessarily true as  $\mathbf{x}_b$  is unlikely to lie in the positive half-space of  $\mathbf{a}_j$ . In contrast, shift vectors for points in the cluster core, converge. Each core member lies in the positive half-space of shift vectors of several other core points. This generates a non-zero probability of transition between nodes in the core, making its constituents nodes in a strongly connected subgraph of  $G$ . Since,  $P$  represents the transition probability matrix

for this digraph  $G$ , taking powers of  $P$  simulates random walks on the graph  $G$ . It is straightforward to see that these random walks move points from the cluster boundary towards the core of the cluster. The preference for points to transition to the cluster boundary disappears after a few iterations.

The row  $\mathbf{p}_i^N$  of the  $P^N$ , represents the probability of a walk starting at  $\mathbf{x}_i$  transitioning to other nodes in the graph over  $N$  steps. Once the walk transitions to the core of the cluster, this transition probability vector begins to converge to a steady state value. This is a direct consequence of the core of the cluster being a strongly connected subgraph. We denote the transition probability matrix, with rows that have converged to a steady state value, as  $P^\epsilon$ . We obtain this matrix by multiplying out the rows  $\mathbf{p}_i$  repeated with  $P$  until, the normed difference in probability distributions between consecutive iterations is less than  $\epsilon$ . We denote this final probability vector as  $\mathbf{p}_i^\epsilon$ . There are two interpretations of the entries in  $\mathbf{p}_i^\epsilon$ : (1) we view,  $j = \arg \max(\mathbf{p}_i^\epsilon)$  as the most likely final destination of a walk starting at  $\mathbf{x}_i$  and use this to perform a connected components labeling. (2) Alternatively,  $\mathbf{p}_i^\epsilon$  may be viewed as a soft assignment of final destinations of a walk originating at  $\mathbf{x}_i$ . We can compare distributions of  $\mathbf{p}^\epsilon$  for different nodes to construct an affinity matrix. We perform spectral clustering on this matrix to give us the final clustering.

## 4 Algorithms

### 4.1 Partitioning by Connected Destinations

Given  $P^\epsilon$ , a straightforward algorithm is to assign each node to its most probable destination. Nodes with the same destination are grouped in the same cluster. However, it is possible that some of the destinations themselves converge on other nodes. Therefore, a simple connected components labeling algorithm is executed to obtain the final labeling. We will refer to this algorithm as Clustering with Shift Vectors (CSV).

### 4.2 Supervised and Unsupervised Spectral Partitioning

Assuming points with similar probability distributions of their final destinations are more likely to belong to the same cluster, we can obtain a similarity matrix for our data by comparing these distributions. Clustering is obtained by spectral analysis of this similarity matrix. Alternatively, the similarity matrix can be constructed based on the initial transition probability distributions  $P$ . We discuss both alternatives here.

Given  $P$ , each row represents a probability mass function for a corresponding node in  $G$  transitioning to other nodes in its influence neighborhood. Let  $\hat{P}$  denote the row-normalized version of the matrix  $P$ . We can define an similarity matrix based on  $\hat{P}$  as:  $A^I = \hat{P}\hat{P}^T$ .

Nodes with preferences to transition to similar parts of the cluster interior have a higher similarity than nodes which transition to other clusters or other parts of the same cluster interior. Consequently, the matrix  $A^I$  is very sparse.



Similarly, given  $P^\epsilon$ , each row represents a probability mass function (PMF) for the final destination of the corresponding point. Points with similar PMF's are more likely to belong to the same cluster (or same part of the cluster) than points with different PMF's. We use this intuition to define a similarity matrix  $A^\epsilon$  as:  $A^\epsilon = \hat{P}^\epsilon \hat{P}^{\epsilon T}$ .

We observed that  $A^\epsilon$  is blockwise dense. Nodes in a cluster usually converge to the same subset of nodes in the core. Therefore, the similarities of their PMF's are likely to be very high.

In the **supervised** setting, the user specifies the number of partitions,  $K$ . We perform a  $K$ -way graph partitioning following the method of [2]: (1) Compute the normalized laplacian  $L = D^{-\frac{1}{2}} A^I D^{-\frac{1}{2}}$  (or  $A^\epsilon$ ). Here  $D$  denotes the degree matrix. (2) Compute the top  $K$  eigenvectors of  $L$  and stack them columnwise in a matrix  $E$ . (3) Normalize rows of  $E$ . (4) Perform  $K$ -means clustering on rows of  $E$  to obtain final clustering. We refer to this algorithm as the Spectral Clustering with Shift Vectors (SCSV- $K$ ).

In the **unsupervised** setting we adopt the eigenvector alignment algorithm proposed in [1] to automatically determine the number of clusters: (1) Given choices for number of clusters  $K_c = \{K_1 \dots K_m\}$  compute the top  $\max(K_c)$  eigenvectors of the normalized laplacian of  $A^I$  (or  $A^\epsilon$ ). (2) For each column subset  $1 : K_c(i)$  of the eigenvector matrix  $E$ , compute the rotation that best aligns this column with the canonical coordinates, by gradient descent. (3) Score the alignment based on the distortion measure [1], to obtain  $C_{K_c(i)}$ . (4) Return the number of clusters as the number of columns with the best alignment score. (5) Stack corresponding columns to form  $E$  and normalize its rows. (6) Return the final clustering as the output of  $K_{best}$ -means algorithm. We refer to this algorithm as Zelnik-Perona Clustering with Shift Vectors (ZPCSV). We preprocess both  $P$  and  $P^\epsilon$  to remove outliers by removing all nodes with zero transition probabilities to other nodes.

## 5 Results

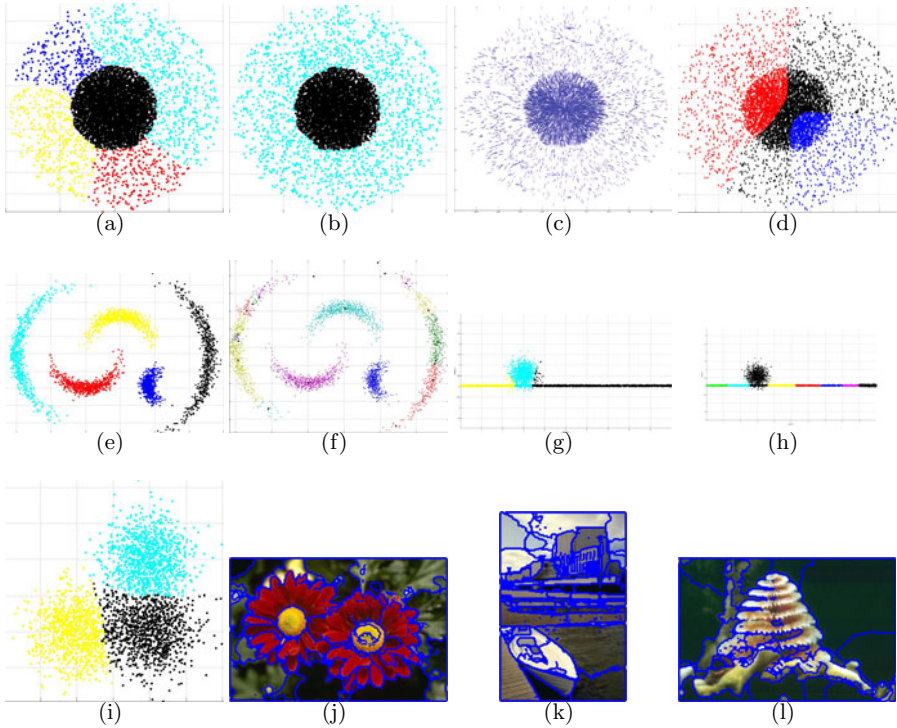
In this section we present the results of experiments with three variants of our algorithm: Clustering with Shift Vectors (CSV), Spectral Clustering with Shift Vectors (SCSV- $K$ ) and Zelnik-Perona Clustering with Shift Vectors (ZPCSV). We first present qualitative results on challenging artificial datasets. We then compare the SCSV- $K$  algorithm against  $K$ -Means(KM), Locally Scaled Spectral Clustering (ls-SC) [1] and Normalized Cuts (NC) [2]. We compare our unsupervised algorithms, CSV and ZPCSV, against Adaptive bandwidth Meanshift (AMS) [3] and the Zelnik-Perona Spectral Clustering (ZPC)<sup>1</sup>.

**Implementation Details:** We do not address the issue of selection of an optimal  $\alpha$  parameter for testing density isotropy, for a dataset. We expect it to

<sup>1</sup> <http://webee.technion.ac.il/~lihi/Demos/SelfTuningClustering.html>

<sup>2</sup> <http://www.cis.upenn.edu/~jshi/software/>

<sup>3</sup> <http://www.caip.rutgers.edu/riul/research/code/AMS/index.html>



**Fig. 4.** (a) Output of our CSV algorithm on overlapping clusters with different densities. (b) Output of the ZPCSV algorithm which determines the right number of clusters based on the criterion described in [11]. (c) Shift vectors computed by our method. (d) Output of Adaptive bandwidth Mean shift on the same data. (e) Our performance on the crescent dataset from [9] and (f) the output of medoid-shift for an arbitrary bandwidth setting. (g) Clusters detected by ZPCSV for a Gaussian overlapping with an elongated uniform cluster from [14]. (h) The output of Zelnik-Perona Clustering on the same data. (i) Output of CSV on three overlapping Gaussian clusters data. (j-l) The output of CSV for color image segmentation. Notice that outliers are detected as isolated pixels within image segments.

be a function of degree of sampling in the data, but this discussion is beyond the scope of this paper. To deal with the diverse datasets in our experiments we computed shift vectors at  $\alpha = \{0.05, 0.025, 0.01, 0.0075, 0.005, 0.0025, 0.001\}$ , for each point. The final shift vector at a point is computed as the vector sum of shift vectors obtained at that point, at each significance level. We used the uniform weighting function  $w\|\cdot\| = 1$  for computing forces (equation [1]), and the entries in the transition probability matrix (equation [5]). We used the triangular weighting kernel in equation [4]. We specified the number of clusters between 1 and 20, for the unsupervised spectral clustering algorithm to evaluate its cost function. We set  $\epsilon = 5e - 4$ . These settings were used for all experiments with artificial and real data.

## 5.1 Artificial Data

The experiments on artificial datasets demonstrate the ability of our approach to cluster (1) both uniform and modal clusters (fig. 4(a)-4(b), 4(i)), (2) multiscale clusters (fig. 4(b)), (3) overlapping clusters of different densities (fig. 4(a)-4(b)) and (4) clusters with arbitrary shapes (fig. 4(e)). (1), (2) and (3) are challenging cases for X-shift based algorithms as shown in figure 4(d). Figure 4(b) shows that ZPCSV, is a reasonable approach to utilizing outputs of our probabilistic shift algorithm to identify the correct number of clusters in data. The corresponding output of CSV is shown in figure 4(a). Figure 4(f) shows that though X-shift based approaches can detect arbitrarily shaped clusters, this too relies on a proper bandwidth setting. In contrast, we use the same parameter settings across all datasets, demonstrating our invariance across a wide variety of data. We compare the outputs of ZPC and ZPCSV on a dataset with a gaussian overlapping an elongated uniform cluster, a challenging dataset from [14]. Self-tuning clustering using the scaled K-NN kernel oversegments the dataset (fig 4(h)). However, the ZPCSV algorithm accurately picks the right number of clusters, while accounting for the density discontinuity that arises when the two distributions overlap (fig 4(g)). This demonstrates that the affinity matrix obtained through probabilistic shift captures local geometry better than the locally-scaled affinity matrix suggested in [1].

## 5.2 Real Data

We use real data to provide quantitative comparisons between our approach and popular algorithms in literature. We use USPS digits, MNIST digits, and datasets from the UCI Machine Learning repository for comparing algorithms. We also demonstrate an application of our algorithm to segment color images (figures 4(j)-4(l)).

**Measuring Clustering Accuracy:** For all our evaluation tasks, we work with data for which the labeling is known. We define clustering accuracy as follows. For each cluster detected, we check the number of unique “ground-truth” labels present. Next we determine the label class with maximum representation within each cluster. The remaining points in the cluster are identified as being wrongly clustered. The clustering error is the percentage of points in the dataset that are assigned to wrong clusters.

**Digits Data:** We use 9268 USPS digits (16×16 images digits 0-9) and 10000 MNIST digits (28×28 images of 0-9) to compare the performance of clustering algorithms. These datasets pose an interesting challenge. The same digit written by different people is likely to be more similar to other digits from the same class, producing distinct clusters. However, some digits have very similar appearances, e.g. 4’s and 9’s, and produce overlapping clusters.

We vectorize the digit images from USPS into 256 and MNIST into 784 dimensions. In the first set of experiments, within each dataset, we gave each of the 45 possible pairs of digits as inputs to the clustering algorithms (e.g. 0’s vs. 1’s).

We report average performance of each algorithm on these 45 pairs in tables [11](#) and [12](#). SCSV-K outperforms all other supervised clustering algorithms on both datasets (table [11](#)). This indicates the affinity matrix constructed using shift vectors produces a more accurate picture of cluster structure. Similarly, ZPCSV outperforms all other unsupervised clustering algorithms. ZPCSV is not restricted to finding 2 clusters in the experiments, therefore it can find multiple clusters within a single digit. However, the median number of clusters returned was 2 for both datasets. CSV shows similar performance however the median number of clusters was 3 for MNIST and 4 for USPS, indicating a tendency to fragment clusters. On average we found 7 outliers (out of 1500-2000 points) per experiment for both datasets. We also experimented with giving all 10 classes simultaneously to the clustering algorithms. Here too SCSV-K outperformed other supervised clustering algorithms. Among unsupervised algorithms CSV performed the best (table [12](#)). Comparatively, ZPCSV performs worse because it finds fewer clusters than there are classes in the data. However, it still beats the original ZPC on both datasets, reinforcing our claim that we construct better affinity matrices using shift vectors. To compare our results against adaptive bandwidth meanshift we varied the initial bandwidth at samples between 10 and 1500 nearest neighbors. We report the best results obtained for their algorithm over this range. We used this best performing bandwidth setting for experiments with all digits. AMS performed worse than all other algorithms compared. Curiously, when we performed K-means clustering on the modes to which individual points in AMS converge, we obtained better results. For example, in the USPS digits clustering task with 10 classes, we obtained an accuracy of 26.52% with K=10, when we post-processed the converged AMS modes using K-means. We attribute this variation in performance to the heuristics employed in merging modes and specifying a minimum cluster size. In contrast we do not invoke heuristics to post process our clustering.

**UCI ML:** We also tested our algorithms on data sets from the UCI Machine Learning Repository(see tables). Our K-way algorithm outperformed other clustering algorithms on most tasks. Interestingly, we noticed that our performance on the SVMGUIDE dataset improved significantly we performed spectral analysis on our the affinity matrix obtained from the initial transition matrix  $P$ , instead of using  $P^c$ . These gains were not significant for other tasks. Although CSV returned 2 clusters, one of them contained over 95% of the points in the dataset. We obtained lower error rates with stricter  $\alpha$  criterion. These results suggest that direct analysis of  $P$  avoids the rare cases where CSV fails to find the right cluster cores. This is because the affinity matrix computed using  $P$  relies on the local correlations of shift vectors, as opposed to the final destinations of the shift. It should be noted that spectral clustering using the affinity matrix obtained from  $P$ , consistently outperforms all other spectral clustering techniques(table [12](#)). From the table it appears that on an unknown dataset CSV would give lower expected error than other methods discussed here.

**Table 1.** Comparison of supervised clustering errors (%) for various datasets. The average error is reported for pairwise experiments with MNIST digits and USPS digits.

Data	KM	NC	ls-SC	SCSV(K)	
				$P^e$	$P$
MNIST(Pairs)	9.04	8.8	8.3	<b>2.7</b>	2.77
MNIST(All)	38.54	80.56	59.49	<b>16.2</b>	17.3
USPS(Pairs)	7.54	5.1	5.13	<b>0.89</b>	0.976
USPS(All)	26.51	30.07	50.1	<b>10.92</b>	18.82
Ionosphere	28.8	10.83	10.82	<b>3.65</b>	<b>3.65</b>
Breast-Cancer	3.95	34.99	34.99	4	<b>3.5</b>
Diabetes	34.9	34.9	<b>34.51</b>	34.85	34.85
SVM-Guide	23.5	12.42	19.61	43.42	<b>5.85</b>

**Table 2.** Comparison of unsupervised clustering errors (%). The numbers in brackets are the number of clusters detected.

Data	AMS	ZPC	ZPCSV		CSV
			$P^e$	$P$	
MNIST(Pairs)	45.2	7.8	<b>1.42</b>	1.62	2.48
MNIST(All)	79.3(9)	80.2(2)	49.87(5)	40.23(7)	<b>17.2(12)</b>
USPS(Pairs)	38.8	3.84	<b>0.913</b>	0.987	0.984
USPS(All)	71.9(9)	69.9(2)	18.75(8)	25.61(7)	<b>4.7(15)</b>
Ionosphere	10.86(3)	10.83(15)	<b>3.65(9)</b>	<b>3.65(10)</b>	<b>3.65(2)</b>
Breast-Cancer	26.35(5)	<b>3.2(7)</b>	4(2)	3.5(2)	3.83(8)
Diabetes	34.37(3)	33.98(2)	<b>33.55(10)</b>	34.85(2)	33.81(4)
SVM-Guide	<b>4.8(12)</b>	9.03(8)	43.42(10)	6.46(6)	43.22(2)

## 6 Conclusions and Contributions

This paper makes three chief contributions. (1) We have introduced a novel scale selection criterion based on density isotropy for the computation of shift vectors. (2) Probabilistic shift using these shift vectors are shown to perform better than X-shift methods on both real and challenging artificial datasets. (3) Affinity matrices computed using these shift vectors are shown to consistently outperform both supervised and unsupervised spectral clustering algorithms. We argue this is a direct consequence of the principled evidence accumulation approach adopted to determine local shift properties for points. One drawback of the probabilistic shift approach is the computational complexity of computing  $P^e$  ( $O(N^3)$ ) by matrix multiplication. In future work we will explore avenues to compute this efficiently or to approximate it.

*Acknowledgement.* The support of the Office of Naval Research under grant N00014-09-1-0017 and the National Science Foundation under grant IIS 08-12188 is gratefully acknowledged.

## References

1. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: *Advances in Neural Information Processing Systems*, vol. 17, pp. 1601–1608 (2004)
2. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: *Advances in NIPS*, vol. 14 (2002)
3. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 888–905 (2000)
4. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 603–619 (2002)
5. Jain, A., Murty, M., Flynn, P.: Data clustering: A review. *Surveys* 31(3), 264–323 (1999)
6. Xu, R., Wunsch, D.: Survey of clustering algorithms. *IEEE Transactions on Neural Networks* 16 (2005)
7. Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory* 21, 32–40 (1975)
8. Koontz, W.L.G., Narendra, P.M., Fukunaga, K.: A graph-theoretic approach to nonparametric cluster analysis. *IEEE Trans. Comput.* 25, 936–944 (1976)
9. Sheikh, Y.A., Khan, E., Kanade, T.: Mode-seeking by medoidshifts. In: *Eleventh IEEE International Conference on Computer Vision* (2007)
10. Shapira, L., Avidan, S., Shamir, A.: Mode-detection via median-shift. In: *2009 IEEE 12th International Conference on Computer Vision*, pp. 1909–1916 (2009)
11. Comaniciu, D., Ramesh, V., Meer, P.: The variable bandwidth mean shift and data-driven scale selection. In: *Proceedings of Eighth IEEE International Conference on Computer Vision, ICCV 2001*, vol. 1, pp. 438–445 (2001)
12. Chung, F.R.K.: *Spectral Graph Theory*. CBMS Regional Conference Series in Mathematics, vol. 92. American Mathematical Society, Providence (1997)
13. Mailla, M., Shi, J.: A random walks view of spectral segmentation. In: *AI and STATISTICS (AISTATS)* (2001)
14. Nadler, B., Galun, M.: Fundamental limitations of spectral clustering. In: *Advances in NIPS*, pp. 1017–1024 (2007)
15. Ahuja, N.: A transform for multiscale image segmentation by integrated edge and region detection. *PAMI* 18, 1211–1235 (1996)
16. Gibbons, J.D.: *Nonparametric Statistical Inference*. Marcel Dekker, New York (1985)
17. Shetty, S., Ahuja, N.: A uniformity criterion and algorithm for data clustering. In: *Proceedings of the 19th ICPR* (2008)

# Depth-Encoded Hough Voting for Joint Object Detection and Shape Recovery

Min Sun<sup>1</sup>, Gary Bradski<sup>2</sup>, Bing-Xin Xu<sup>1</sup>, and Silvio Savarese<sup>1</sup>

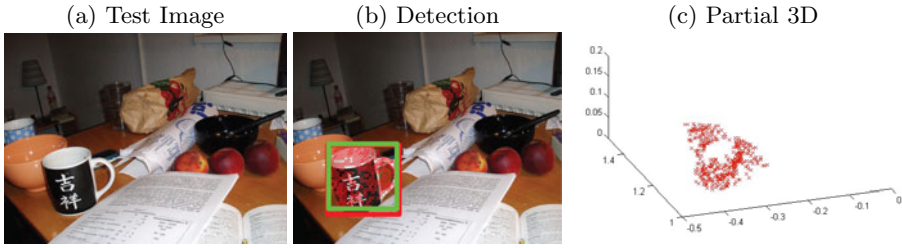
<sup>1</sup> Electrical and Computer Engineering  
University of Michigan, Ann Arbor, USA  
{sunmin,xbx}@umich.edu, silvo@eecs.umich.edu  
<sup>2</sup> Willow Garage, Menlo Park, CA, USA  
bradski@willowgarage.com

**Abstract.** Detecting objects, estimating their pose and recovering 3D shape information are critical problems in many vision and robotics applications. This paper addresses the above needs by proposing a new method called DEHV - Depth-Encoded Hough Voting detection scheme. Inspired by the Hough voting scheme introduced in [13], DEHV incorporates depth information into the process of learning distributions of image features (patches) representing an object category. DEHV takes advantage of the interplay between the scale of each object patch in the image and its distance (depth) from the corresponding physical patch attached to the 3D object. DEHV jointly detects objects, infers their categories, estimates their pose, and infers/decodes objects depth maps from either a single image (when no depth maps are available in testing) or a single image augmented with depth map (when this is available in testing). Extensive quantitative and qualitative experimental analysis on existing datasets [6,9,22] and a newly proposed 3D table-top object category dataset shows that our DEHV scheme obtains competitive detection and pose estimation results as well as convincing 3D shape reconstruction from just one single uncalibrated image. Finally, we demonstrate that our technique can be successfully employed as a key building block in two application scenarios (highly accurate 6 degrees of freedom (6 DOF) pose estimation and 3D object modeling).

## 1 Introduction

Detecting objects and estimating their geometric properties are crucial problems in many application domains such as robotics, autonomous navigation, high-level visual scene understanding, activity recognition, and object modeling. For instance, if one wants to design a robotic system for grasping and manipulating objects, it is of paramount importance to encode the ability to accurately estimate object orientation (pose) from the camera view point as well as recover structural properties such as its 3D shape. This information will help the robotic arm grasp the object at the right location and successfully interact with it.

This paper addresses the above needs, and tackles the following challenges:  
i) Learn models of object categories by combining view specific depth maps



**Fig. 1.** Illustration of key steps in our method. Given a single (previously) unseen testing image (panel a), our DEHV (Depth-Encoded Hough Voting-based) scheme is used to detect objects (panel b). Ground truth bounding box is shown in red. Our detection is shown in green. The centers of the image patches which cast votes for the object location are shown in red crosses. During detection, our method simultaneously infers object depth maps of the detected object (panel c). This allows the estimation of the partial 3D shape of the object from a single image!



**Fig. 2.** Point clouds (green) from a 3D model is registered to the inferred partial 3D point cloud (red) by DEHV (a). This allows us to achieve an accurate 6 DOF pose estimation (b) and realistic 3D object modeling (c).

along with the associated 2D image of objects in the same class from different vantage points. We demonstrate that combining imagery with 3D information helps build richer models of object categories that can in turn make detection and pose estimation more accurate. ii) Design a coherent and principled scheme for detecting objects and estimating their pose from either just a single image (when no depth maps are available in testing) (Fig. 1(b)), or a single image augmented with depth maps (when these are available in testing). In the latter case, 3D information can be conveniently used by the detection scheme to make detection and pose estimation more robust than in the single image case. iii) Have our detection scheme recover the 3D structure of the object from just a single uncalibrated image (when no 3D depth maps are available in testing) (Fig. 1(c)) and without having seen the object instance during training.

Inspired by implicit shape model (ISM) [13], our method is based on a new generalized Hough voting-based scheme [2] that incorporates depth information into the process of learning distributions of object image patches that are compatible with the underlying object location (shape) in the image plane. We call our scheme *DEHV - Depth-Encoded Hough Voting scheme* (Sec. 3). DEHV addresses the intrinsic weaknesses of existing Hough voting schemes [13, 10, 16, 17] where errors in estimating the scale of each image object patch directly affects the ability of the algorithm to cast consistent votes for the object existence. To



resolve this ambiguity, we take advantage of the interplay between the scale of each object patch in the image and its distance (depth) from the corresponding physical patch attached to the 3D object, and specifically use the fact that objects (or object parts) that are closer to the camera result in image patches with larger scales. Depth is encoded in training by using available depth maps of the object from a number of view points. At recognition time, DEHV is applied to detect objects (Fig. 1(b)) and simultaneously infer/decode depths given hypotheses of detected objects (Fig. 1(c)). This process allows the reinforcement of the existence of an object even if a depth map is not available in testing. If depth maps are available in testing, the additional information can be used to further validate if a given detection hypothesis is correct or not. As a by-product of the ability of DEHV to infer/decode depth at recognition time, we can estimate the location in 3D of each image patch involved in the voting, and thus recover the partial 3D shape of the object. Critically, depth decoding can be achieved even if just a single test image is provided. Extensive experimental analysis on a number of public datasets (including car Pascal VOC07 [6], mug ETHZ Shape [9], mouse and stapler 3D object dataset [21]) as well as a newly created in-house dataset (comprising 3 object categories) are used to validate our claims (Sec. 4). Experiments with the in-house dataset demonstrate that our DEHV scheme: i) achieves better detection rates (compared to the traditional Hough voting scheme); further improvement is observed when depth maps are available in testing; ii) produces convincing 3D reconstructions from single images; the accuracy of such reconstructions have been qualitatively assessed with respect to ground truth depth maps. Experiments with public datasets demonstrate that our DEHV successfully scales to different types of categories and works in challenging conditions (severe background clutter, occlusions). DEHV achieves state of the art detection results on several categories in [6,9], and competitive pose estimation results on [21]). Finally, we show anecdotal results demonstrating that DEHV is capable to produce convincing 3D reconstructions from single uncalibrated images from [6,9,21] in Fig. 12.

We demonstrated the utility of DEHV in two applications (Sec. 4.3): i) Robot object manipulation: we show that DEHV enables accurate 6 DOF pose estimation (Fig. 2(b)); ii) 3D object modeling: we show that DEHV enables the design of a system for obtaining eye catching 3D objects models from just one single image (Fig. 2(c));

## 2 Previous Work

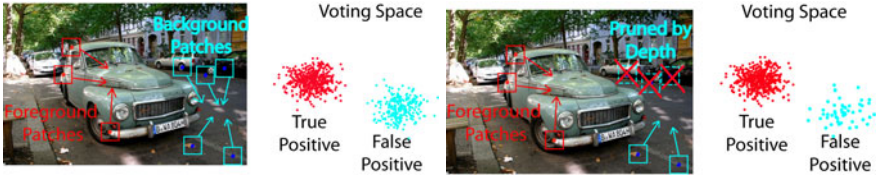
In the last decade, the vision community has made substantial progress addressing the problem of object categorization from 2D images. While most of the work has focussed on representing objects as 2D models [4,13,8] or collections of 2D models [23], very few methods have tried to combine in a principled way the appearance information that is captured by images and the intrinsic 3D structure representative of an object category. Works by [25,21,22] have proposed

solutions for modeling the way how 2D local object features (or parts) and their relationship vary in the image as the camera view point changes. Other works [11,27,14,1] propose hybrid models where reconstructed 3D object models are augmented with features or parts capturing diagnostic appearance. Few of them (except [26] for objects) have demonstrated and evaluated the ability to recover 3D shape information from a single query image. However, instead of using image patches to transfer meta-data (like depth) to the testing instance as in [26], 3D information is directly encoded into our model during training. Other works propose to address the problem of detecting and estimating geometrical properties of single object instances [12,19,18,15]; while accurate pose estimation and 3D object reconstruction are demonstrated, these methods cannot be easily extended to incorporate intra-class variability so as to detect and reconstruct object categories. Unlike our work, these techniques also require that the objects have significant interior texture to carry out geometric registration. Other approaches assume that additional information about the object is available in both training and testing (videos, 3D range data) [20,5]. Besides relying on more expensive hardware platforms, these approaches tend to achieve high detection accuracy and pose estimation, but fail when the additional 3D data is either partially or completely unavailable.

### 3 Depth-Encoded Hough Voting

In recognition techniques based on hough voting [2] the main idea is to represent the object as a collection of parts (patches) and have each part to cast votes in a discrete voting-space. Each vote corresponds to a hypothesis of object location  $x$  and class  $O$ . The object is identified by the conglomeration of votes in the voting space  $V(O, x)$ .  $V(O, x)$  is typically defined as the sum of independent votes  $p(O, x, f_j, s_j, l_j)$  from each part  $j$ , where  $l_j$  is the location of the part,  $s_j$  is the scale of the part, and  $f_j$  is the part appearance.

Previously proposed methods [13,10,16,17] differ mainly by the mechanism for selecting good parts. For example, parts may be either selected by an interest point detector [13,16], or densely sampled across many scales and locations [10]; and the quality of the part can be learned by estimating the probability [13] that the part is good or discriminatively trained using different types of classifiers [16,10]. In this paper, we propose a novel method that uses 3D depth information to guide the part selection process. As a result, our constructed voting space  $V(O, x|D)$ , which accumulates votes for different object classes  $O$  at location  $x$ , depends on the corresponding depth information  $D$  of the image. Intuitively, any confusing part that is selected at a wrong scale can be pruned out by using depth information. This allows us to select parts which are consistent with the object physical scale. It is clear that depending on whether object is closer or further, or depending on the actual 3D object shape, the way how each patch votes will change (Fig. 3).



**Fig. 3.** Left panel shows that patches associated to the actual object parts (red boxes) will vote for the correct object hypothesis (red dots) in the voting space on the right. However, parts from the background or other instances (cyan boxes) will cast confusing votes and create a false object hypothesis (green dots) in the voting space. Right panel shows that given depth information, the patches selected in a wrong scale can be easily pruned. As a result, the false positive hypothesis will be supported by less votes.

In detail, we define  $V(O, x|D)$  as the sum of individual probabilities over all observed images patches at location  $l_j$  and for all possible scales  $s_j$ , i.e.,

$$\begin{aligned}
 V(O, x|D) &= \sum_j \int p(O, x, f_j, s_j, l_j|d_j) ds_j \\
 &= \sum_j \int p(O, x|f_j, s_j, l_j, d_j)p(f_j|s_j, l_j, d_j)p(s_j|l_j, d_j)P(l_j|d_j) ds_j \quad (1)
 \end{aligned}$$

where the summation over  $j$  aggregates the evidence from individual patch location, and the integral over  $s_j$  marginalizes out the uncertainty in scale for each image patch. Since  $f_j$  is calculated deterministically from observation at location  $l_j$  with scale  $s_j$ , and we assume  $p(l_j|d_j)$  is uniformly distributed given depth, we obtain:

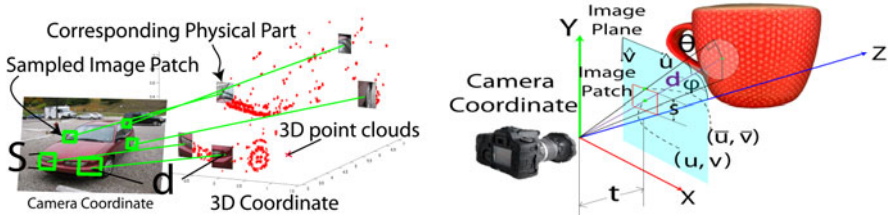
$$\begin{aligned}
 V(O, x|D) &\propto \sum_j \int p(O, x|f_j, s_j, l_j, d_j)p(s_j|l_j, d_j)ds_j \\
 &= \sum_{j,i} \int p(O, x|C_i, s_j, l_j, d_j)p(C_i|f_j)p(s_j|l_j, d_j)ds_j \quad (2)
 \end{aligned}$$

Here we introduce codebook entry  $C_j$ , matched by feature  $f_j$ , into the framework, so that the quality of a patch selected will be related to which codeword it is matched to. Noting that  $C_j$  is calculated only using  $f_j$  and not the location  $l_j$ , scale  $s_j$ , and depth  $d_j$ , we simplify  $p(C_j|f_j, s_j, l_j, d_j)$  into  $p(C_j|f_j)$ . And by assuming that  $p(O, x|\cdot)$  does not depend on  $f_j$  given  $C_j$ , we simplify  $p(O, x|C_j, f_j, s_j, l_j, d_j)$  into  $p(O, x|C_j, s_j, l_j, d_j)$ .

Finally, we decompose  $p(O, x|\cdot)$  into  $p(O|\cdot)$  and  $p(x|\cdot)$  as follows:

$$V(O, x|D) \propto \sum_{j,i} \int p(x|O, C_i, s_j, l_j, d_j)p(O|C_i, s_j, l_j, d_j)p(C_i|f_j)p(s_j|l_j, d_j) ds_j$$

**Scale to depth mapping.** We design our method so as to specifically selects image patches that tightly enclose a sphere with a fix radius  $r$  in 3D during



**Fig. 4.** Illustration of depth to scale mapping. Right panel illustrates the concept of depth to scale mapping. Training under the assumption that an image patch (green box) tightly encloses the physical 3D part with a fix size, our method deterministically selects patches given the patch center  $l$ , 3D information of the image, and focal length  $t$ . During testing, given the selected image patches on the object, our method directly infers the location of the corresponding physical parts and obtains the 3D shape of the object. Left Panel illustrates the physical interpretation of Eq. 3. Under the assumption that image patch (red bounding box) tightly encloses the 3D sphere with radius  $r$ , the patch scale  $s$  is directly related to the depth  $d$  given camera focal length  $t$  and the center  $l = (u, v)$  of the image patch. Notice that this is a simplified illustration where the patch center is on the  $yz$  plane. This figure is best viewed in color.

training. As a result, our model enforces a 1-to-1 mapping  $m$  between scale  $s$  and depth  $d$ . This way, given the 3D information, our method deterministically select the scale of the patch at each location  $l$ , and given the selected patches, our method can infer the underlying 3D information (Fig 4). In detail, given the camera focal length  $t$ , the corresponding scale  $s$  at location  $l = (u, v)$  can be computed as  $s = m(d, l)$  and the depth  $d$  can be inferred from  $d = m^{-1}(s, l)$ . The mapping  $m$  obeys the following relations:

$$\begin{aligned}
 s &= 2(\bar{v} - v); \quad \bar{v} = \tan(\theta + \phi)t; \quad \theta = \arcsin\left(\frac{r}{d_{yz}}\right); \quad \phi = \arctan\left(\frac{v}{t}\right) \\
 d_{yz} &= \frac{d\sqrt{t^2 + v^2}}{\sqrt{u^2 + v^2 + t^2}} : \text{d projected onto } yz \text{ plane}
 \end{aligned}
 \tag{3}$$

Hence,  $p(s|l, d) = \delta(s - m(d, l))$ . Moreover, using the fact that there is a 1-to-1 mapping between  $s$  and  $d$ , probabilities  $p(x|\cdot)$  and  $p(O|\cdot)$  are independent to  $d$  given  $s$ . As a result, only scale  $s$  is directly influenced by depth.

In the case when depth is unknown,  $p(s|l, d)$  becomes a uniform distribution over all possible scales. Our model needs to search through the scale space to find patches with correct scales. This will be used to detect the object and simultaneously infer the depth  $d = m^{-1}(s, l)$ . Hence, the underlying 3D shape of the object will be recovered.

**Random forest codebook.** In order to utilize dense depth map or infer dense reconstruction of an object, we use random forest to efficiently map features  $f$  into codeword  $C$  (similar to [10]) so that we can evaluate patches densely distributed over the object. Moreover, random forest is discriminatively trained

to select salient parts. Since feature  $f$  deterministically maps to  $C^i$  given the  $i_{th}$  random tree, the voting score  $V(O.x|D)$  becomes:

$$V(O, x|D) \propto \sum_{j,i} \int p(x|O, C^i(f_j), s_j, l_j)p(O|C^i(f_j))p(s_j|l_j, d_j) ds_j \quad (4)$$

where the summation over  $i$  aggregates the discriminative strength of different trees. In section 3.1, we describe how the distributions of  $p(x|O, C^i(f_j), s_j, l_j)$  and  $p(O|C^i(f_j))$  are learned given training data, so that each patch  $j$  knows where to cast votes during recognition.

### 3.1 Training the Model

We assume that for a number of training object instances, the 3D reconstruction  $D$  of the object is available. This corresponds to having available the distance (depth) of each image object patch from its physical location in 3D. Our goal is to learn the distributions of location  $p(x|\cdot)$  and object class  $p(O|\cdot)$ , and the mapping of  $C^i(f)$ . Here we define location  $x$  of an object as a bounding box with center position  $q$ , height  $h$ , and aspect ratio  $a$ . We sample each image patch centered at location  $l$  and select the scale  $s = m(l, d)$ . Then the feature  $f$  is extracted from the patch  $(l, s)$ . When the image patch comes from a foreground object, we cache: 1) the information of the relative voting direction  $b$  as  $\frac{q-l}{s}$ ; 2) the relative object-height/patch-scale ratio  $w$  as  $\frac{h}{s}$ ; 3) the object aspect ratio  $a$ . Then, we use both the foreground patches (positive examples) and background patches (negative examples) to train a random forest to obtain the mapping  $C^i(f)$ .  $p(O|C)$  is estimated by counting the frequency that patches of  $O$  falls in the codebook entry  $C$ .  $p(x|O, C, s, l)$  can be evaluated given the cached information  $\{v, w, a\}$  as follows:

$$p(x|O, C, s, l) \propto \sum_{j \in g(O, C)} \delta(q - b_j \cdot s + l, h - w_j \cdot s, a - a_j) \quad (5)$$

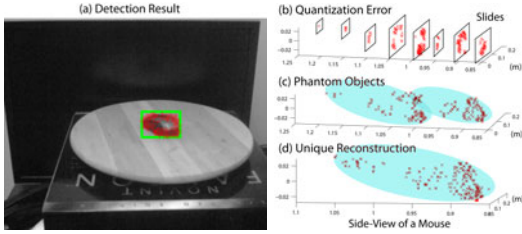
where  $g(O, C)$  is a set of patches from  $O$  mapped to codebook entry  $C$ .

### 3.2 Recognition and 3D Reconstruction

**Recognition when depth is available.** It is straightforward to use the model when 3D information is observed during recognition. Since the uncertainty of scale is removed, Eq. 4 becomes

$$V(O, x|D) \propto \sum_{j,i} p(x|O, C^i(f_j), m(l_j, d_j), l_j)p(O|C^i(f_j)) \quad (6)$$

Since  $s_j = m(l_j, d_j)$  is a single value at each location  $j$ , the system can detect objects more efficiently by computing less features and counting less votes. Moreover, patches selected using local appearance at a wrong scale can be pruned out to reduce hallucination of objects (Fig. 3).



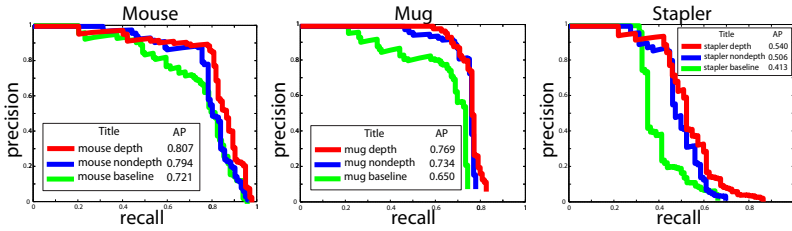
**Fig. 5.** A typical detection result in (a) shows object hypothesis bounding box (green box) and patches (red crosses) vote for the hypothesis. A naive reconstruction suffers from quantization error (b) and phantom objects (c). Our algorithm overcomes these issues and obtains (d)

**Recognition when depth is not available.** When no 3D information is available during recognition,  $p(s_j|l_j, d_j)$  becomes a uniform distribution over the entire scale space. Since there is no closed form solution of integral over  $s_j$ , we propose to discretize the space into a finite number of scales  $S$  so that Eq. 4 can be approximated by  $V(O, x|D) \propto \sum_{j,i} \sum_{s_j \in S} p(x|O, C^i(f_j), s_j, l_j)p(O|C^i(f_j))$

**Decoding 3D information.** Once we obtain a detection hypothesis  $(x, O)$  (Green box in Fig. 5(a)) corresponding to a peak in the voting space  $V$ , the patches that have cast votes for a given hypothesis can be identified (Red cross in Fig. 5(a)). Since the depth information is encoded by the scale  $s$  and position  $l$  of each image patch, we apply Eq 3 in a reverse fashion to infer/decode depths from scales. The reconstruction, however, is affected by a number of issues: i) **Quantization error:** The fact that scale space is discretized into a finite set of scales, implies that the depths  $d$  that we obtained are also discretized. As a result, we observe the reconstructed point clouds as slices of the true object (See Fig. 5(b)). We propose to use the height of the object hypothesis  $h$  and the specific object-height/patch-scale ratio  $w$  to recover the continuous scale  $\hat{s} = h/w$ . Notice that since  $w$  is not discretized,  $\hat{s}$  is also not discretized. Hence, we recover the reconstruction of an object as a continuum of 3D points (See Fig. 5(c)). ii) **Phantom objects:** The strength and robustness of our voting-based method comes from the ability to aggregate pieces of information from different training instances. As a result, the reconstruction may contain multiple phantom objects since image patches could resemble those coming from different training instances with slightly different intrinsic scales. Notice that the phantom objects phenomenon reflects the uncertainty of the scale of the object in an object categorical model. In order to construct a unique shape of the detected object instance, we calculate the relative object height in 3D with respect to a selected reference instance to normalize the inferred depth. Using this method, we recover a unique 3D shape of the detected object.

## 4 Evaluation

We evaluated our DEHV algorithm on several datasets. The training settings were as follows. For each training image, we randomly sample 100 image patches



**Fig. 6.** Object localization results are shown as precision recall curves evaluated using PASCAL VOC protocol. (Green curve) Result using standard ISM model (baseline). (Blue curve) Result using DEHV with no depth information during testing. (Red curve) Result using DEHV with partial depth information during testing. Notice the consistent improvement of average precision (AP) compared to the baseline hough voting.

from object instances and 500 image patches from background regions. The scale of the patch size from the corresponding object instance is determined by its (known) depth (Fig. 4). At the end, 10 random trees (Sec. 3.1) are trained using the sampled foreground and background patches for each dataset. For all experiment, we use a Hog-like feature introduced in [10]. During detection, our method treats each discrete viewpoint as a different class  $O$ .

#### 4.1 Exp.I: System Analysis on a Novel 3D Table-Top Object Dataset

Due to the lack of datasets comprising both images and 3D depth maps of set of generic object categories, we propose a new 3D table-top object category dataset collected on a robot platform. The dataset contains three common table-top object categories: mice, mugs, and staplers, each with 10 object instances. We arrange these objects in two different sets for the purpose of object localization and pose estimation evaluation. The object localization dataset (Table-Top-Local) contains 200 images with the number of object ranging from 2 to 6 object instances per image in a clutter office environment. The object pose estimation dataset (Table-Top-Pose) contains 480 images where each object instance is captured under 16 different poses (8 angles and 2 heights). For both settings, each image comes with depth information collected using a structure-light stereo camera. Please see the author’s project page (<http://www.eecs.umich.edu/~sunmin>) for more information about the dataset.

We evaluate our method under 3 different training and testing conditions, which are 1) standard ISM model trained and tested without depth, 2) DEHV trained with depth but tested without depth, and 3) DEHV trained and tested with depth. We show that the knowledge of 3D information helps in terms of object localization (Fig. 6), and pose estimation (Fig. 7). Moreover, we evaluate our method’s ability to infer depth from just a single 2D image. Given the ground truth focal length of the camera, we evaluate the absolute depth error for the inferred partial point clouds in table. 1-Left Column. Notice that our

(a) Standard ISM Average Perf.=49.4%											(b) DEHV w/o depth Average Perf.=51.0%											(c) DEHV w/ depth Average Perf.=63.0%											
front	.27	.00	.09	.09	.45	.09					f	.28	.00	.08	.24	.00	.16	.00				f	.12	.00	.00	.05	.83	.00	.00	.00			
front-left	.46	.15	.04	.15	.15	.04					fl	.00	.01	.00	.11	.00	.05	.21	.00				fl	.11	.42	.11	.00	.32	.05	.00	.00		
left	.00	.00	.64	.00	.00	.04	.28	.04			l	.00	.08	.79	.08	.00	.00	.04	.00				l	.00	.04	.77	.00	.12	.00	.08	.00		
left-back	.00	.07	.07	.61	.00	.00	.21	.04			lb	.00	.04	.12	.77	.00	.00	.08	.00				lb	.00	.00	.08	.73	.12	.00	.00	.08		
back	.00	.25	.00	.00	.08	.00	.58	.08			b	.16	.04	.00	.12	.56	.00	.12	.00				b	.09	.00	.05	.05	.82	.00	.00	.00		
back-right	.00	.09	.00	.00	.00	.00	.67	.35	.00		br	.00	.08	.00	.12	.00	.62	.15	.04				br	.04	.04	.04	.00	.39	.65	.04	.00		
right	.00	.00	.17	.00	.00	.00	.79	.04			r	.00	.04	.12	.12	.00	.00	.72	.00				r	.00	.14	.07	.14	.00	.61	.04	.00		
right-front	.00	.11	.00	.07	.00	.00	.48	.33			rf	.00	.00	.10	.45	.00	.00	.25	.20				rf	.04	.00	.00	.25	.21	.00	.04	.96		
	f	fl	l	lb	b	br	r	rf			f	fl	l	lb	b	br	r	rf				f	fl	l	lb	b	br	r	rf				

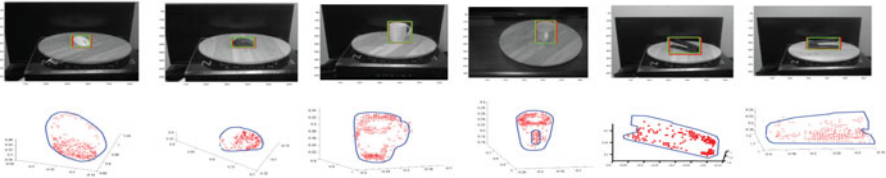
**Fig. 7.** Pose estimation results averaged across three categories. The average accuracy increases when more 3D information is available. And knowing depths in both training and testing sets gives the best performance.

errors are always lower than the baseline errors<sup>1</sup>. We also evaluate the relative depth errors<sup>2</sup> reported in table. 1-Right Column when the exact focal length is unknown. Object detection examples and inferred 3D point clouds are shown in Fig. 8.

Table 1	Abs. Depth in (m) (known focal length)	Rel. Depth (unknown focal length)
	Sparse/Baseline	Sparse/Baseline
Mouse	0.0145/0.0255	0.0173/0.0308
Mug	0.0176/0.0228	0.0201/0.0263
Stapler	0.0094/0.0240	0.0114/0.0298

Table 2. pose estimation performance on 3D object dataset[21]

DEHV stapler	DEHV mouse	Savarese et al. '08 [22]	Farhadi et al. '09 [7]
75.0	73.5	64.78	78.16



**Fig. 8.** Example of object detections (Top) and inferred 3D point clouds (Bottom). The inferred point clouds preserve the detailed structure of the objects, like the handle of mug. Object contours are overlaid on top of the image to improve the readers understanding. Please refer to the author’s project page for a better visualization.

## 4.2 Exp.II: Comparison on Three Challenging Datasets

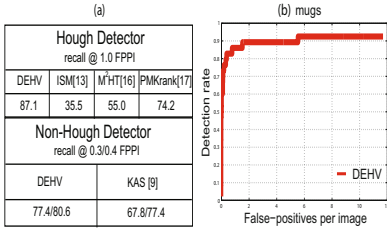
In order to demonstrate that DEHV generalizes well on other publicly available datasets, we compare our results with state-of-the-art object detectors on a subset of object categories from the ETHZ shape dataset, 3D object dataset, and Pascal 2007 dataset. Notice that all of these datasets contain 2D images only. Therefore, training of DEHV is performed using the 2D images from these public available dataset and the depth maps available from the 3D table-top dataset and our own set of 3D reconstruction of cars<sup>3</sup>.

<sup>1</sup> It is computed assuming each depth is equal to the median of the depths of the inferred partial point clouds.

<sup>2</sup>  $\frac{\|d-\hat{d}\|}{d}$  where  $d$  is the ground truth depth, and  $\hat{d}$  is the estimated depth. And  $\hat{d}$  is scaled so that  $d$  and  $\hat{d}$  have the same median.

<sup>3</sup> Notice that only depth is used from our own dataset.





**Fig. 9.** Performance on the mug category of ETHZ shape dataset [9]. (a-Top) Performance comparison with other pure Hough voting methods (M<sup>2</sup>HT) [16] and (PMK rank) [17]. (a-Bottom) Performance comparison between state-of-the-art non-hough voting methods [9]. (b) Detection Rate vs. FPPI of DEHV.

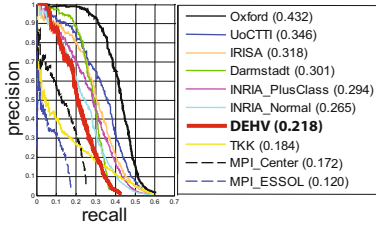
**ETHZ Shape Dataset.** We test our method on the Mug category of the ETHZ Shape dataset. It contains 48 positive images with mugs and 207 negative images with a mixture of apple logos, bottles, giraffes, mugs, and swans. Following the experiment setup in [9], we use 24 positive images and an equal number of negative images for training. We further match the 24 mugs with the mugs in 3D table-top object dataset to transfer the depth maps to the matched object instances so that we obtain augmented depth for positive training images. All the remaining 207 images in the ETHZ Shape dataset are used for testing.

The table in Fig. 9(a)-top shows the comparison of our method with the standard ISM and two state-of-the-art pure voting-based methods at 1.0 False-Positive-Per-Image (FPPI). Our DEHV method (recall 83.0 at 1 FPPI) significantly outperforms Max-Margin Hough Voting (M<sup>2</sup>HT) [16] (recall 55 at 1 FPPI) and pyramid match kernel ranking (PMK ranking) [17] (recall 74.2 at 1 FPPI). The table in Fig. 9(a)-bottom shows that our method is comparable to state-of-the-art non-voting-based method KAS [9]. Note that these results are not including a second stage verification step which would naturally boost up performance. The recall vs (FPPI) curve of our method is shown in Fig. 9(b).

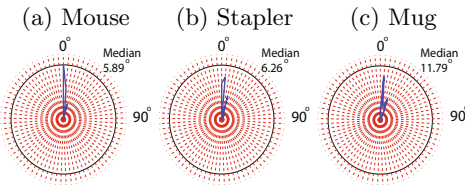
**3D object dataset.** We test our method on the mouse and stapler categories of the 3D object dataset [21,22], where each category contains 10 object instances observed under 8 angles, 3 heights, and 2 scales. We adapt the same experimental settings as [21,22] with additional depth information from the first 5 instances of the 3D table-top object dataset to train our DEHV models. The pose estimation performance of our method is shown in table.2. It is superior than [22] and comparable to [7] (which primarily focuses on pose estimation only).

**Pascal VOC 2007 Dataset.** We tested our method on the car category of the Pascal VOC 2007 challenge dataset [6], and report the localization performance. Unfortunately PASCAL does not contain depth maps. Thus, in order to train DEHV with 3D information, we collect a 3D car dataset containing 5 car instances observed from 8 viewpoints, and use Bundler [24] to obtain its 3D reconstruction. We match 254 car instances<sup>4</sup> in the training set of Pascal 2007 dataset to the instances in 3D car dataset and associate depth maps to these 254 Pascal training images. This way the 254 positive images can be associated to a rough depth value. Finally, both 254 positive Pascal training images and

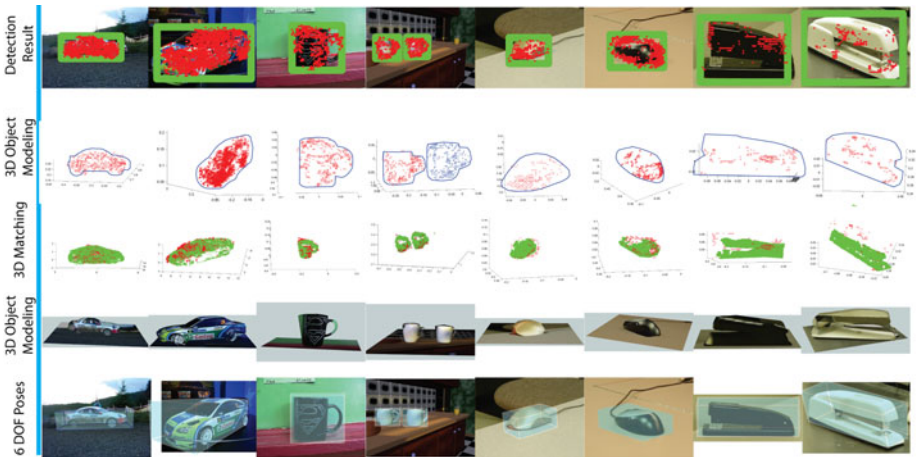
<sup>4</sup> 254 cars is a subset of the 1261 positive images in the PASCAL training set. The subset is selected if they are easy to match with the 3D car dataset.



**Fig. 10.** Object Localization result using PASCAL VOC07 dataset. The precision-recall generated by our method (red) is compared with the results of 2007 challenge [6]-Oxford, [6]-UoCTTI, [6]-IRISA, [6]-Darmstadt, [6]-INRIAPlusClass, [6]-INRIANormal, [6]-TKK, [6]-MPICenter, [6]-MPIESSOL.



**Fig. 11.** Circular histograms of 6 DOF error in degree, where 6 DOF error is defined as the angle (in degree) between the ground truth and estimated object orientation (ex: a normalized 3D vector pointing from the center to the front of an object.)



**Fig. 12.** Examples of the complete 3D object inference process using the testing images from Pascal VOC07 [6], ETHZ Shape [9], and 3D object dataset [21]. This figure should be viewed in color. **Row 1** Detection results (green box) overlaid with image patch centers (red cross) which cast the votes. **Row 2** Inferred 3D point clouds (red dots), given the detection results. **Row 3** 3D registration results, where red indicates the inferred partial point clouds and green indicates the visible parts of the 3D CAD model. **Row 4** 3D Object modeling using the 3D CAD models and estimated 3D pose of the objects. Notice that the supporting plane in 3D object modeling are manually added. **Row 5** Visualizations of the estimated 6 DOF poses. (See author’s project page for 3D visualization.)

the remaining 4250 negative images are used to train our DEHV detector. We obtain reasonably good detection performance (Average Precision 0.218) even though we trained with fewer positive images (Fig. 10). Detection examples and inferred objects 3D shape are shown in Fig. 12.

### 4.3 Applications: 6 DOF Pose Estimation and 3D Object Modeling

DEHV detects object classes, estimates a rough pose, and infers a partial reconstruction of the detected object. In order to robustly recover the accurate 6 DOF pose and the complete 3D shape of the object, we propose to register the inferred partial 3D point cloud (Fig. 1(c)) to a set of complete 3D CAD models<sup>5</sup>. Having estimated pose during detection allows us to highly reduce the complexity of this registration process. A modified ICP algorithm [3] is used for registration. Quantitative evaluation of 6 DOF pose estimation are shown in Fig. 11. We also obtain a full 3D object model by texture mapping the 2D image onto the 3D CAD model. Anecdotal results are reported in the 5<sup>th</sup> row of figure 12.

## 5 Conclusion

We proposed a new detection scheme called DEHV which can successfully detect objects, estimate their pose from either a single 2D image or a 2D image combined with depth information. Most importantly, we demonstrated that DEHV is capable of recover the 3D shape of object categories from just one single uncalibrated image.

**Acknowledgments.** We acknowledge the support of NSF (Grant CNS 0931474) and the Gigascale Systems Research Center, one of six research centers funded under the Focus Center Research Program (FCRP), a Semiconductor Research Corporation Entity, and Willow Garage, Inc. for collecting the 3D table-top object category dataset.

## References

1. Arie-Nachimson, M., Basri, R.: Constructing implicit 3d shape models for pose estimation. In: ICCV (2009)
2. Ballard, D.H.: Generalizing the hough transform to detect arbitrary shapes. In: Pattern Recognition (1981)
3. Besl, P.J., McKay, H.D.: A method for registration of 3-d shapes. IEEE Trans. PAMI 14(2), 239–256 (1992)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
5. Deselaers, T., Criminisi, A., Winn, J., Agarwal, A.: Incorporating on-demand stereo for real time recognition. In: CVPR (2007)
6. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge, VOC 2007 Results (2007)
7. Farhadi, A., Tabrizi, M.K., Endres, I., Forsyth, D.: A latent model of discriminative aspect. In: ICCV (2009)

---

<sup>5</sup> The models are collected only for registration usage.

8. Fergus, R., Perona, P., Zisserman, A.: A sparse object category model for efficient learning and exhaustive recognition. In: CVPR (2005)
9. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of adjacent contour segments for object detection. *IEEE Trans. PAMI* 30(1), 36–51 (2008)
10. Gall, J., Lempitsky, V.: Class-specific hough forests for object detection. In: CVPR (2009)
11. Hoeim, D., Rother, C., Winn, J.: 3d layoutcrf for multi-view object class recognition and segmentation. In: CVPR (2007)
12. Huttenlocher, D.P., Ullman, S.: Recognizing solid objects by alignment with an image. *IJCV* 5(2), 195–212 (1990)
13. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: ECCV workshop on statistical learning in computer vision (2004)
14. Liebelt, J., Schmid, C., Schertler, K.: Viewpoint-independent object class detection using 3d feature maps. In: CVPR (2008)
15. Lowe, D.G.: Local feature view clustering for 3d object recognition. In: CVPR (2001)
16. Maji, S., Malik, J.: Object detection using a max-margin hough transform. In: CVPR (2009)
17. Ommer, B., Malik, J.: Multi-scale object detection by clustering lines. In: ICCV (2009)
18. Romea, A.C., Berenson, D., Srinivasa, S., Ferguson, D.: Object recognition and full pose registration from a single image for robotic manipulation. In: ICRA (2009)
19. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3D object modeling and recognition using affine-invariant patches and multi-view spatial constraints. In: CVPR (2003)
20. Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M.: Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in human environments. In: IROS (2009)
21. Savarese, S., Fei-Fei, L.: 3D generic object categorization, localization and pose estimation. In: ICCV (2007)
22. Savarese, S., Fei-Fei, L.: View synthesis for recognizing unseen poses of object classes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 602–615. Springer, Heidelberg (2008)
23. Schneiderman, H., Kanade, T.: A statistical approach to 3D object detection applied to faces and cars. In: CVPR (2000)
24. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. In: SIGGRAPH (2006)
25. Su, H., Sun, M., Fei-Fei, L., Savarese, S.: Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In: ICCV (2009)
26. Thomas, A., Ferrari, V., Leibe, B., Tuytelaars, T., Van Gool, L.: Using multi-view recognition and meta-data annotation to guide a robot’s attention. *Int. J. Rob. Res.* (2009)
27. Yan, P., Khan, D., Shah, M.: 3d model based object class detection in an arbitrary view. In: ICCV (2007)

# Shape Analysis of Planar Objects with Arbitrary Topologies Using Conformal Geometry

Lok Ming Lui<sup>1</sup>, Wei Zeng<sup>2,3</sup>, Shing-Tung Yau<sup>1</sup>, and Xianfeng Gu<sup>3</sup>

<sup>1</sup> Department of Mathematics, Harvard University, Cambridge, MA, USA

<sup>2</sup> Department of Computer Science, Wayne State University, Detroit, MI, USA

<sup>3</sup> Department of Computer Science, SUNY Stony Brook, Stony Brook, NY, USA

**Abstract.** The study of 2D shapes is a central problem in the field of computer vision. In 2D shape analysis, classification and recognition of objects from their observed silhouettes are extremely crucial and yet difficult. It usually involves an efficient representation of 2D shape space with natural metric, so that its mathematical structure can be used for further analysis. Although significant progress has been made for the study of 2D simply-connected shapes, very few works have been done on the study of 2D objects with *arbitrary topologies*. In this work, we propose a representation of general 2D domains with arbitrary topologies using *conformal geometry*. A natural metric can be defined on the proposed representation space, which gives a metric to measure dissimilarities between objects. The main idea is to map the exterior and interior of the domain conformally to unit disks and circle domains, using holomorphic 1-forms. A set of diffeomorphisms from the unit circle  $S^1$  to itself can be obtained, which together with the conformal modules are used to define the shape signature. We prove mathematically that our proposed signature uniquely represents shapes with arbitrary topologies. We also introduce a reconstruction algorithm to obtain shapes from their signatures. This completes our framework and allows us to move back and forth between shapes and signatures. Experiments show the efficacy of our proposed algorithm as a stable shape representation scheme.

## 1 Introduction

Shape analysis of objects from their observed silhouettes is important for many computer vision applications, such as classification, recognition and image retrieval. In order to perform shape analysis effectively, it is necessary to have an efficient shape representation and a robust metric measuring shape dissimilarity.

Recently, many different representations for 2D shapes and various measures of dissimilarity between them have been proposed. For example, Zhu et al. [1] proposed the representation of shapes using their medial axis and compare their skeletal graphs through a branch and bound strategy. Liu et al. [2] used shape axis trees to represent shapes, which are defined by the locus of midpoints of optimally corresponding boundary points. Belongie et al. [3] proposed to represent and match 2D shapes for object recognition, based on the shape context and the

Hungarian method. Mokhtarian [4] introduced a multi-scale, curvature-based shape representation technique for planar curves, which is especially suitable for recognition of a noisy curve. Besides, various statistical models for shape representation were also proposed by different research groups [5,6,7]. These approaches provide a simple way to represent shapes with finite dimensional spaces, although they cannot capture all the variability of shapes. Yang et al. [8] proposed a signal representation called the Schwarz representation and applied it to shape matching problems. Lee et al. [9] proposed to represent curves using harmonic embedding through their complete silhouettes. Lipman et al. [10] proposed to detect shape dissimilarities up to isometry using conformal densities. Their works focus on simply-connected domains. Zeng et al. [11] presented to match and register 3D multiply-connected domains using holomorphic differentials. Zeng et al. [12] analyzed 3D surfaces based on conformal modules. Their shape index can only determine shapes up to conformal deformations. Mumford et al. [13] proposed a conformal approach to model simple closed curves which captured subtle variability of shapes up to scaling and translation. They also introduced a natural metric, called the Weil-Petersson metric, on the proposed representation space.

Most of the above methods work only on simple closed curves and generally cannot deal with multiply-connected objects. In real world applications, objects from their observed silhouettes are usually multiply-connected domains (i.e. domains with holes in the interior). In order to analyze such kind of shapes effectively, it is necessary to develop an algorithm which can deal with multiply-connected domains. This motivates us to look for a good representation, which is equipped with a natural metric, to model planar objects of arbitrary topologies.

In this paper, we extend Mumford's conformal approach [13], which models 2D simply-connected domains, to represent multiply-connected shapes. Mumford's approach provides an effective way to represent 2D simple curves and capture their subtle differences. To extend it to multiply-connected shapes, the key idea of our method is to map the exterior and interior of the domain conformally to unit disks and punctual disks, using holomorphic 1-forms. A set of diffeomorphisms from the unit circle  $\mathbb{S}^1$  to itself can be obtained, which together with the conformal modules are used to define the shape signature. Our proposed signature uniquely represents shapes with arbitrary topologies up to scaling and translation. We also introduce a reconstruction algorithm to obtain shapes from their signatures. This completes our framework and allows us to move back and forth between shapes and signatures. The proposed representation space inherits a natural metric that can be used to measure dissimilarity between shapes.

## 2 Theoretically Background

In this section, we briefly introduce the theoretical foundations necessary for the current work. For more details, we refer readers to the classical books [14,15].

### 2.1 Beltrami Equation

Consider a complex valued function  $\phi : \mathbb{C} \rightarrow \mathbb{C}$  maps the  $z$ -plane to the  $w$ -plane, where  $z = x + iy$ ,  $w = u + iv$ . The *complex partial derivative* is defined as:  $\frac{\partial}{\partial z} := \frac{1}{2}(\frac{\partial}{\partial x} - i\frac{\partial}{\partial y})$ ,  $\frac{\partial}{\partial \bar{z}} = \frac{1}{2}(\frac{\partial}{\partial x} + i\frac{\partial}{\partial y})$ . The *Beltrami equation* for  $\phi$  is defined by:  $\frac{\partial \phi}{\partial \bar{z}} = \mu(z)\frac{\partial \phi}{\partial z}$ , where  $\mu$  is called the *Beltrami coefficient*. If  $\mu$  is zero, then  $\phi$  is called a *holomorphic* or *conformal mapping*. Otherwise, if  $\|\mu\|_\infty < 1$ , then  $\phi$  is called a *quasiconformal mapping*. Given a compact simply-connected domain  $\Omega$  in  $\mathbb{C}$  and a Beltrami coefficient  $\mu$  with  $\|\mu\|_\infty < 1$ . There is always a quasiconformal mapping from  $\Omega$  to the unit disk  $\mathbb{D}$  which satisfies the Beltrami equation in the distribution sense [14].

### 2.2 Conformal Module

Suppose  $\Omega_1$  and  $\Omega_2$  are planar domains. We say  $\Omega_1$  and  $\Omega_2$  are *conformally equivalent* if there is a biholomorphic diffeomorphism between them. All planar domains can be classified by the conformal equivalence relation. Each conformal equivalence class shares the same *conformal invariants*, the so-called *conformal module*. The conformal module is one of the key component for us to define the unique shape signature.

Suppose  $\Omega$  is a compact domain on the complex plane  $\mathbb{C}$ . If  $\Omega$  has a single boundary component, then it is called a *simply-connected domain*. Every simply connected domain can be mapped to the unit disk conformally and all such kind of mappings differ by a *Möbius transformation*:  $z \rightarrow e^{i\theta} \frac{z-z_0}{1-\bar{z}_0z}$ .

Suppose  $\Omega$  has multiple boundary components  $\partial\Omega = \gamma_0 - \gamma_1 - \gamma_2 \cdots \gamma_n$ , where  $\gamma_0$  represents the exterior boundary component, then  $\Omega$  is called a *multiply-connected domain*. A *circle domain* is a unit disk with circular holes. Two circle domains are conformally equivalent, if and only if they differ by a Möbius transformation. It turns out every multiply-connected domain can be conformally mapped to a circle domain, as described in the following theorem.

**Theorem 1 (Riemann Mapping for Multiply-Connected Domain).** *If  $\Omega$  is a multiply-connected domain, then there exists a conformal mapping  $\phi : \Omega \rightarrow D$ , where  $D$  is a circle domain. Such kind of mappings differ by Möbius transformations.*

Therefore, each multiply-connected domain is conformally equivalent to a circle domain. The conformal module for a circle domain is represented as the centers and radii of inner boundary circles. All simply-connected domains are conformally equivalent. The topological annulus requires 1 parameter to represent the conformal module. In general case, because there are  $n > 1$  inner circles, and the Möbius transformation group is 3 dimensional, therefore the conformal module requires  $3n - 3$  parameters. We denote the conformal module of  $\Omega$  as  $Mod(\Omega)$ . Fix  $n$ , all conformal equivalence classes form a  $3n - 3$  Riemannian manifold, the *Teichmüller space*. The conformal module can be treated as the Teichmüller coordinates. The Weil-Petersson metric [13] is a Riemannian metric for Teichmüller space, which induces negative sectional curvature, therefore, the geodesic between arbitrary two points is unique.

### 2.3 Holomorphic Differentials

In order to compute the conformal modules, one needs to find the holomorphic differential forms on the multiply-connected domain. A *differential 1-form* on a planar domain  $\omega$  is defined as  $\tau = f(x, y)dx + g(x, y)dy$ , where  $f, g$  are smooth functions. The *Hodge star* operator acting on a differential 1-form gives the *conjugate differential 1-form*  $*\tau = -g(x, y)dx + f(x, y)dy$ . Intuitively, the conjugate 1-form  $*\tau$  is obtained by rotating  $\tau$  by a right angle everywhere.

A *holomorphic 1-form* consists of a pair of conjugate harmonic 1-forms  $\omega = \tau + i*\tau = \phi(z)dz$ , where  $\phi(z)$  is a holomorphic function. We further requires that either  $\tau$  or  $*\tau$  is orthogonal to all the boundaries. All holomorphic 1-forms form a group (with real coefficients), denoted as  $\mathbb{H}(\Omega)$ . A basis of  $\mathbb{H}(\Omega)$  is given by:  $\{\omega_1, \omega_2, \dots, \omega_n\}$ , such that  $\int_{\gamma_j} \omega_i = \delta_i^j$ , where  $\delta_i^j$  is the Kronecker symbol.

By integrating the holomorphic 1-forms, one can construct the conformal *circular slit map*, whose existence is guaranteed by the following theorem.

**Theorem 2 (Circular Slit Map).** *Suppose  $\Omega$  is a multiply connected domain with more than one boundary components, then there exists a conformal mapping  $\phi : \Omega \rightarrow \mathbb{C}$ , such that  $\gamma_0, \gamma_1$  are mapped to concentric circles,  $\gamma_k$ 's are mapped to concentric circular slits. All such kind of mappings differ by a rotation.*

### 2.4 Conformal Welding

This work is built on *conformal welding*, which is constructed as follows. Suppose  $\Gamma = \{\gamma_0, \gamma_1, \dots, \gamma_k\}$  is a set of non-intersecting smooth closed curves on the complex plane.  $\Gamma$  segments the plane to a set of connected components  $\{\Omega_0, \Omega_1, \dots, \Omega_s\}$ , each segment  $\Omega_i$  is a multiply-connected domain. We assume  $\Omega_0$  contains the infinity point,  $p \notin \Omega_0$ . By using a Möbius transformation  $\phi(z) = \frac{1}{z-p}$ ,  $p$  is mapped to  $\infty$ ,  $\Omega_0$  is mapped to a compact domain. Replace  $\Omega_0$  by  $\phi(\Omega_0)$ . Construct  $\phi_k : \Omega_k \rightarrow \mathbb{D}_k$  to map each segment  $\Omega_k$  to a circle domain  $\mathbb{D}_k$ ,  $0 \leq k \leq s$ . Assume  $\gamma_i \in \Gamma = \Omega_j \cap \Omega_k$ , then  $\phi_j(\gamma_i)$  is a circular boundary on the circle domain  $\mathbb{D}_j$ ,  $\phi_k(\gamma_i)$  is another circle on  $\mathbb{D}_k$ . Let  $f_i|_{\mathbb{S}^1} := \phi_j \circ \phi_k^{-1}|_{\mathbb{S}^1} : \mathbb{S}^1 \rightarrow \mathbb{S}^1$  be the diffeomorphism from the circle to itself, which is called the *signature* of  $\gamma_i$ .

**Definition 1 (Signature of a Family of Loops).** *The signature of a family non-intersecting closed planar curves  $\Gamma = \{\gamma_0, \gamma_1, \dots, \gamma_k\}$  is defined as:  $S(\Gamma) := \{f_0, f_1, \dots, f_k\} \cup \{Mod(\mathbb{D}_0), Mod(\mathbb{D}_1), \dots, Mod(\mathbb{D}_s)\}$ .*

The following main theorem plays the fundamental role for the current work.

**Theorem 3 (Main Theorem).** *The family of smooth planar closed curves  $\Gamma$  is determined by its signature  $S(\Gamma)$ , unique up to a Möbius transformation of the Riemann sphere  $\mathbb{C} \cup \{\infty\}$ .*



Note that if a circle domain  $\mathbb{D}_k$  is disk, its conformal module can be omitted from the signature. The Möbius transformation of the Riemann sphere is given by  $(az + b)/(cz + d)$ , where  $ad - bc = 1, a, b, c, d \in \mathbb{C}$ . The proof of Theorem 3 can be found in the Appendix.

The theorem states that the proposed signature determine shapes up to a Möbius transformation. We can further do a normalization that fixes  $\infty$  to  $\infty$  and that the differential carries the real positive axis at  $\infty$  to the real positive axis at  $\infty$ , as in Mumford’s paper [13]. The signature can then determine the shapes uniquely up to translation and scaling.

The shape signature  $S(\Gamma)$  gives us a complete representation for the space of shapes. It inherits a natural metric. Given two shapes  $\Gamma_1$  and  $\Gamma_2$ . Let  $S(\Gamma_i) := \{f_0^i, f_1^i, \dots, f_k^i\} \cup \{Mod(\mathbb{D}_0^i), Mod(\mathbb{D}_1^i), \dots, Mod(\mathbb{D}_s^i)\}$  ( $i = 1, 2$ ). We can define a metric  $d(S(\Gamma_1), S(\Gamma_2))$  between the two shape signatures using the natural metric in the Teichmuller space, such as the Weil-Petersson metric [13].

Besides, our signature is stable under geometric noise. Our algorithm depends on conformal maps from shapes to circle domains using holomorphic 1-forms. The computation of 1-forms is equivalent to solving an elliptic PDE, which is stable under the perturbation of boundary conditions. On the other hand, in theory, the change of topology will cause the change of conformal structures. Hence, our algorithm is sensitive to topological noise. In practice, after extracting the contours, we filter out the ones with length less than the threshold, which are treated as topological noise.

### 3 Algorithm

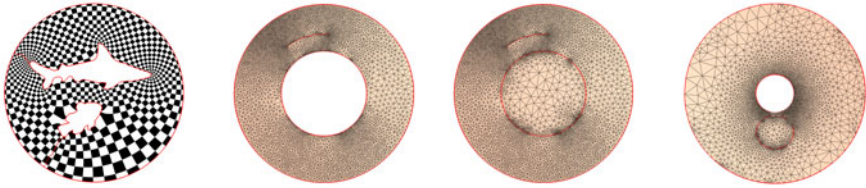
In this section, we describe our proposed algorithm in detail. Here, we assume a planar domain  $\Omega$  is with  $n$  inner boundary components, Let the boundary of the mesh be  $\partial\Omega = \gamma_0 - \gamma_1 \dots - \gamma_n$ , represented as a triangular mesh. We use  $v_i$  to denote a vertex,  $[v_i, v_j]$  denote an edge,  $[v_i, v_j, v_k]$  denote face. The angle at vertex  $v_i$  in triangle  $[v_i, v_j, v_k]$  is denoted as  $\theta_{jk}^i$ . The angle structure of the mesh is defined as the set:  $A(\Omega) := \{\theta_{jk}^i, \theta_{ij}^k, \theta_{ki}^j | [v_i, v_j, v_k] \in \Omega\}$ . In this work, all the following computations completely depend on the angle structure.

#### 3.1 Shape Signatures of Planar Domains with Arbitrary Topologies

We describe the algorithm to compute the signature of  $\Omega$  with  $n$  inner boundary components. The inner boundaries decompose  $\Omega$  into several sub-domains  $\Omega_k$ . The algorithm consists of two main steps, as follows:

- 1. Compute the conformal maps from  $\Omega_k$  to circle domains  $D_k$ ;
- 2. Compute the conformal modules for each sub-domain  $\Omega_k$  and the signature  $f_{ij}$  for each boundary.

**Step 1:** *Conformal maps from  $\Omega_k$  to circle domains  $D_k$ .* The conformal parameterization of  $\Omega_k$  can be obtained easily by computing the circular slit map and performing the Koebe’s iteration. Detailed algorithm can be found in [16].



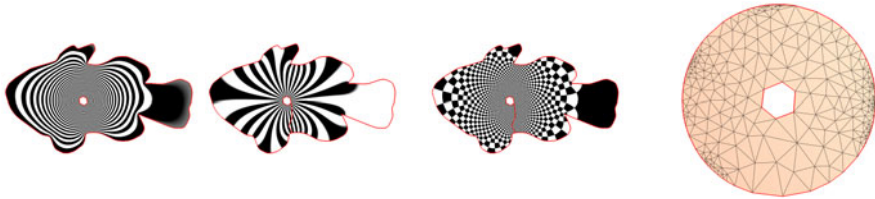
(a) Holomorphic 1-form (b) Circular slit map (c) Fill the inner hole (d) Circular map of (c)

**Fig. 1.** Circular slit map

*Circular slit map:* The circular slit map can be obtained by finding a holomorphic 1-form  $\omega$ , such that

$$Img(\int_{\gamma_0} \omega) = 2\pi, \quad Img(\int_{\gamma_1} \omega) = -2\pi, \quad im.g(\int_{\gamma_k} \omega) = 0, \quad 2 \leq k \leq n. \quad (1)$$

To solve Equation 1, we first compute the basis for the holomorphic 1-form group.  $\omega$  is then a linear combination of the basis  $\omega = \sum_{k=1}^n \lambda_k \omega_k$ , the coefficients  $\{\lambda_k\}$  can be calculated by solving the linear system 1. The circular slit map is given by  $\phi(p) = exp(\int_q^p \omega)$ ,  $\forall p \in \Omega$ , where  $q$  is a base point, and the integration path is arbitrarily chosen in  $\Omega$ . Figure 1 shows the circular slit map of a 2-hole planar domain.



(a) Exact form (b) Closed form (c) Holomorphic form (d) Conformal mapping

**Fig. 2.** Conformal mapping for a simply connected domain by puncturing a small hole in the center

If  $\Omega$  is a simply-connected domain (topological disk), we compute the conformal mapping to map it to the unit disk in the following way. First, we punch a small hole in the domain, and treat it as a topological annulus. Then we use circular slit map to map the punched annulus to the canonical annulus. By shrinking the size of the punched hole, the circular slit mappings converge to the conformal mapping. Figure 2 shows such an example.

*Hole filling:* After computing the circular slit map, the planar domain is mapped to the planar annulus with concentric circular slits.  $\gamma_0$  is the unit circle,  $\gamma_1$  is the inner circle,  $\gamma_k$ 's are slits,  $2 \leq k \leq n$ . We use Delaunay triangulation to generate a disk  $D_1$  bounded by  $\gamma_1$ ,  $\partial D_1 = \gamma_1$ , and glue  $\Omega$  with  $D_1$  along  $\gamma_1$ ,  $\Omega_1 := \Omega \cup_{\gamma_1} D_1$ . We then use circular slit map again to map  $\Omega_1$ , such that

$\gamma_2$  is opened to a circle. We compute a disk  $D_2$  bounded by  $\gamma_2$ , glue  $\Omega_1$  and  $D_2$  to get  $\Omega_2$ . By repeating circular slit map, at the step  $k$ ,  $\gamma_k$  is opened to a circle. We compute a circular disk  $D_k$  bounded by  $\gamma_k$ , and glue  $\Omega_{k-1}$  with  $D_k$ ,  $\Omega_k = \Omega_{k-1} \cup_{\gamma_k} D_k$ . Eventually, we can fill all the holes to get  $\Omega_n$ . All the disks  $D_k$  in  $\Omega_n$  are not exactly circular.

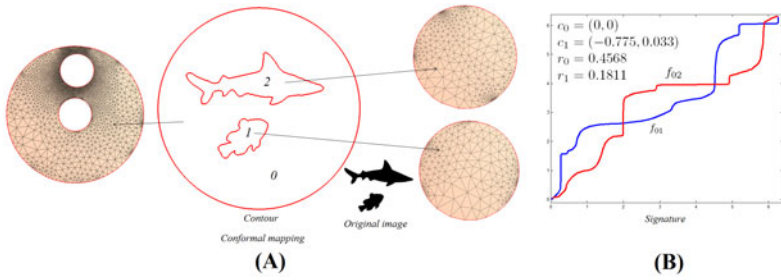
*Koebe’s iteration:* By Koebe’s iteration, all the boundary components become rounder and rounder. Basically, each time, we choose a disk  $D_k$ . The complement of  $D_k$  on  $\Omega_n$  is a doubly-connected domain. We map the complement to the canonical planar annulus, then  $\gamma_k$  becomes a circle. We recompute the disk  $D_k$  bounded by the updated  $\gamma_k$ , and glue the annulus with the updated  $D_k$ . After this iteration,  $\gamma_k$  becomes a circle. Then we choose another disk  $D_j$ , and repeat this process to make  $\gamma_j$  a circle. This will destroy the perfectness of the circular shape of  $\gamma_k$ . But by repeating this process, all the  $\gamma_k$ ’s become rounder and rounder, and eventually converge to perfect circles. The convergence is exponentially fast. Detailed proof can be found in [15].

**Step 2:** *Computing conformal modules and signatures  $f_{ij}$  on boundaries.* After the conformal parameterization of  $\Omega_k$  to the circle domain is computed, we can compute their conformal modules and also the signature  $f_{ij}$  on each boundary. The conformal modules together with  $\{f_{ij}\}$  give the complete signature  $S(\Gamma)$ . We demonstrate the process for computing  $S(\Gamma)$  with a double fish image as shown in Figure 3. Given the original image, we first perform image segmentation to get the binary image, then calculate the contours of the objects in the image. The contour of each fish is shown in the figure. For simplicity, we treat the outermost boundary of the image as the unit circle. Then all the contours segment the image to planar domains  $\Omega_0, \Omega_1, \Omega_2$ . We map each planar segment to a circle domain.  $\Omega_0$  is mapped to a disk  $D_0$  with two circular holes. The centers and radii  $(c_0, r_0)$  and  $(c_1, r_1)$  form the conformal module of  $\Omega_0$ . Also,  $\Omega_1$  and  $\Omega_2$  are mapped to the unit disks  $D_1, D_2$  respectively. We denote the conformal maps of  $\Omega_i$  by  $\Phi_i : \Omega_i \rightarrow D_i$ . The contour of the small fish are mapped to the boundary of  $D_1$  and one inner boundary of  $D_0$ , the signature is given by  $f_{01} := \Phi_1 \circ \Phi_0^{-1}$ , which is shown in frame (B) as the blue curve. Similarly, the signature  $f_{02}$  of the contour of the shark can also be computed. The signature of both fish contours is given by  $S(\Gamma) = \{c_0, c_1, r_0, r_1, f_{01}, f_{02}\}$ .

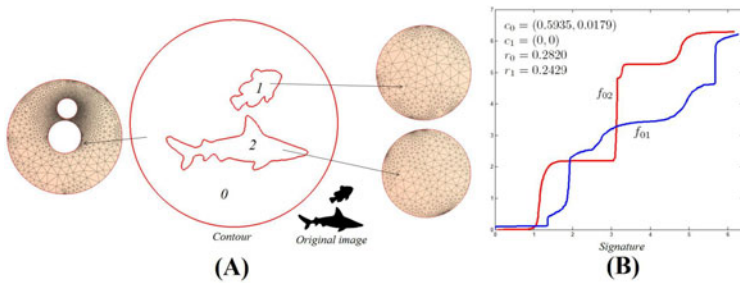
### 3.2 Reconstruction of Shapes from Signatures

Suppose  $\Omega$  has  $n$  contours, then with  $n + 1$  segments. The signature is given by the conformal modules  $\{Mod(D_k), 0 \leq k \leq n\}$  and automorphisms of circles  $f_{ij}$ .

First, we construct circle domains  $D_k$ ’s directly from their modules  $Mod(D_k)$ ’s. We tessellate the circular boundaries of each  $D_k$  and use Delaunay triangulation to triangulate  $D_k$ . Then, we combinatorially glue the triangular mesh  $D_i$  and  $D_j$  by  $f_{ij}$ . Suppose the boundary circle  $\gamma_i \in \partial D_i$  corresponds to  $\gamma_j \in D_j$ ,  $f_{ij} : \gamma_i \rightarrow \gamma_j$ . For each vertex  $v_i \in \gamma_i$ , we insert  $f_{ij}(v_i)$  to  $\gamma_j$ , vice versa, for each vertex  $v_j \in \gamma_j$ , we insert  $f_{ij}^{-1}(v_j)$  to  $\gamma_i$ . Then we use constrained



**Fig. 3.** Signature. Each segment is mapped to a circle domain. The conformal modules (centers and radii of inner circles) of the circle domains and the diffeomorphisms of the circles define the signature.



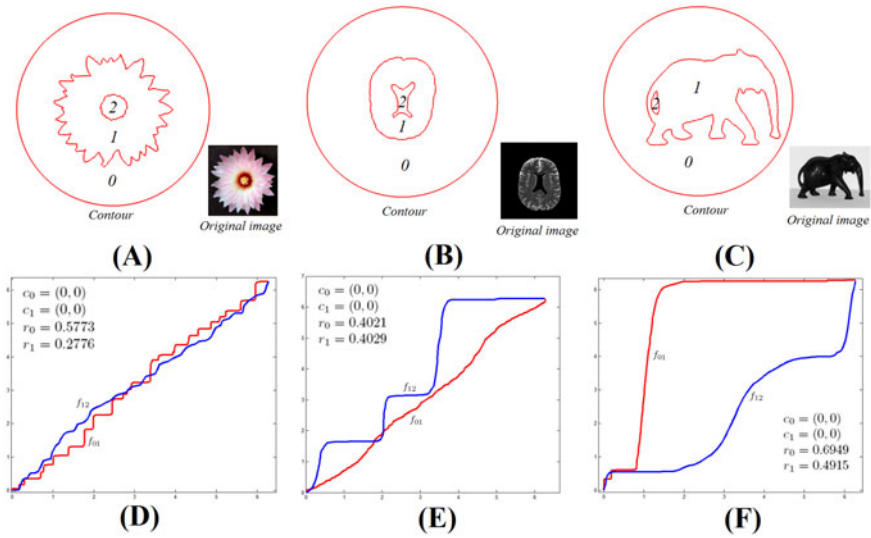
**Fig. 4.** The shark image with spatial changes in the positions of the two fishes. The shape signature can effectively capture spatial changes of objects in the image (compared to Figure 3).

Delaunay triangulation to refine the triangulation of  $D_i$  and  $D_j$ . Therefore the refined triangle mesh  $D_i$  and  $D_j$  can be combinatorially glued through  $\gamma_i$  and  $\gamma_j$ . We repeat this process for all  $f_{ij}$ 's, to obtain a combinatorial triangle mesh, denoted as  $D$ .

In the whole algorithm pipeline, all the computations solely depend on the angle structure. We define the angle structure of  $D$  as:  $A(D) = \cup_{k=0}^n A(D_k)$ .

Then we compute a conformal mapping  $\phi$  from  $D$  to the unit disk using the angle structure  $A(D)$ . The image  $\phi(D)$  differs from the original image by a Möbius transformation. This can be further removed by specifying three vertices on the outer boundary circle.

Suppose in the original image, the positions of three boundary vertices  $\{v_0, v_1, v_2\}$  are  $\{w_0, w_1, w_2\}$ , and their positions in  $\phi(D)$  are  $\{z_0, z_1, z_2\}$ . We need compute a unique Möbius transformation  $\rho$ , such that  $\rho(z_k) = w_k$ . First, we maps the unit disk to the upper half plane by  $h(z) = \frac{z-i}{iz-1}$ . Then on the upper half plane, we map  $\{h(z_0), h(z_1), h(z_2)\}$  to  $\{0, 1, \infty\}$  by  $\sigma_1(z) = \frac{z-h(z_0)}{z-h(z_2)} \frac{h(z_1)-h(z_2)}{h(z_1)-h(z_0)}$ . Similarly, we construct  $\sigma_2(z)$ , that maps  $\{h(w_0), h(w_1), h(w_2)\}$  to  $\{0, 1, \infty\}$ . The the composition map  $\sigma = h^{-1} \circ \sigma_2^{-1} \circ \sigma_1 \circ h$  is the desired Möbius transformation,



**Fig. 5.** Shape signatures of different images with 2 boundaries and 2 levels

which is called *normalization map*. Therefore,  $\tau \circ \phi$  maps  $D$  to the unit disk, which reconstructs the contours from the signature.

## 4 Experimental Results

We implement our proposed algorithm using generic C++ on windows XP platform, with Intel Duo CPU 2.33 GHz, 3.98 G RAM. The numerical systems are solved using Matlab C++ library. The contour extraction is obtained by using the OpenCV library. The computational time for our algorithm is shown in Table 1. In general, both the signature calculation and reconstruction take less than 1 minute to compute, even on complicated domains.

**Table 1.** Computational time (second)

Model	# of contours	# of vertex	# of faces	Signature	Reconstruction
Cat	3	5247	10236	19 s	10 s
TwoCats	6	5969	11680	29 s	7 s
Ameba	2	9094	17930	8 s	12 s
Fishes	2	5978	11716	23 s	8 s
NewFishes	2	7519	14780	24 s	9 s
Elephant	2	11968	23678	17 s	-
Brain	2	8211	16164	11 s	-
Wolf	3	8451	16644	47 s	-

### A. Shape Representation of Multiply-Connected Domains

Figure 4 (A) shows another double fishes image with spatial changes in the positions of the two fishes, compared with that in Figure 3. The big shark and small fish interchanged their positions. The shape signature of the image is plotted in (B), which is quite different from the shape signature in Figure 3 (see red and blue curves). In other words, our shape signature can effectively capture spatial changes of objects in the image, which can be potentially used for the purpose of image understanding. Figure 5 shows the shape signatures of 3 different images with 3 boundaries and 2 levels (levels = number of punctual disks needed for conformal parameterizations). (A) shows the shape signature of the flower image. Note that the fluctuating pattern of the outer boundary of the flower is effectively captured by  $f_{01}$  (the red curve). (B) and (C) shows the shape signatures of the brain and elephant images respectively. The three different images have very different shape signatures, meaning that our shape representation can effectively be used for classifying shapes. We also computed the shape signatures on more complicated images. Figure 6 (A) shows a wolf image with 3 boundaries and 1 level. The exterior and interior of the domain are conformally mapped to the unit disk and punctual disk. The conformal domains consist of one punctual disk with 3 inner disks removed. So, the conformal modules consist of 3 centers and 3 radii, as shown in (A). The diffeomorphisms of the unit circle on each boundary are also plotted. (B) shows the shape signature of the Mickey Mouse image with 3 boundaries and 2 levels. The conformal domain consists of two punctual disks. So, the conformal modules again consist of 3 centers and radii. The conformal modules together with the diffeomorphisms of the unit circle are plotted. Figure 7 shows an image with two cats. It consists of 6 boundaries with 2 levels. The conformal modules consist of 3 punctual disks with 3 holes removed. Hence, the conformal modules consist of 6 centers and 6 radii. The shape signatures are plotted in (B) and (C). (B) shows the signature for the outer level whereas (C) shows the signature of the inner level. Experimental results on these complicated images demonstrate the efficacy of our shape representation method.

### B. Reconstruction of Shapes From Their Shape Signatures

Figure 8 shows the reconstruction of the shark image from its shape signature. The reconstructed image closely resembles to the original image, except some very tiny details are missing. The zoomed views show that the reconstructed ones are smoother, and lose the sharp corners. It shows our algorithm can effectively reconstruct shapes from their signature. We also tested our reconstruction algorithm on images with 2 levels. Figure 9 shows the Ameba image with 2 boundaries and 2 levels. The conformal domains consist of two punctual disks, each has one hole removed. The conformal modules consist of two centers and two radii. The shape signature is plotted in (B). We reconstruct the image from its shape signature in (C), which is very close to the original image. We also tested the algorithm on a more complicated example. Figure 10 shows a cat image with 3 boundaries and 2 levels. As we can see in (A), the original contour of the image is a little bit noisy. We computed the shape signature of the

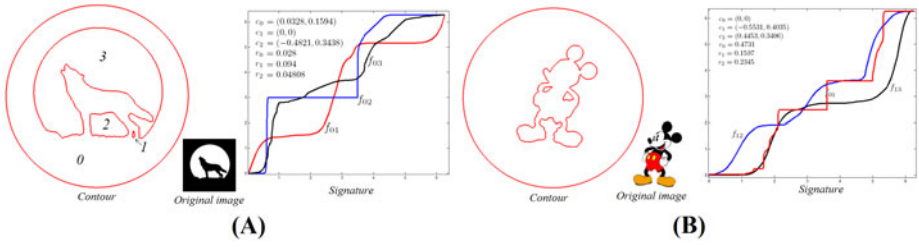


Fig. 6. Shape signatures of different images with 3 boundaries and 2 levels

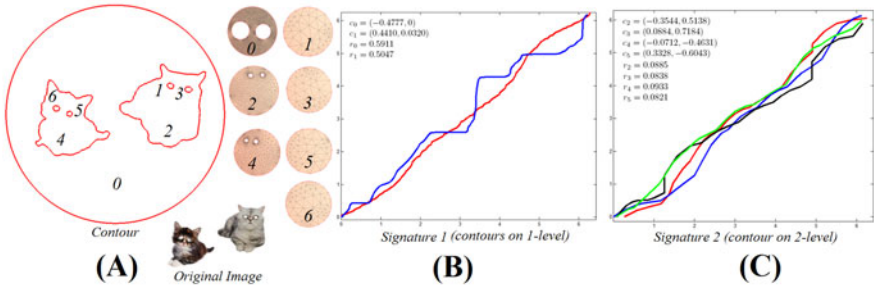
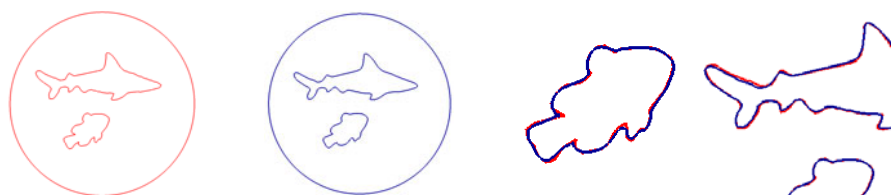


Fig. 7. Shape signatures of another image of cats with 6 boundaries and 2 levels

image, which is shown in (B). In (C), we show the reconstructed image from its shape signature. Again, the reconstructed image is very close to the original one, although the original noisy contours are smoothed out a little bit. Finally, we studied the numerical error of our reconstruction scheme. Table 2 shows the distance between the original and reconstructed contours of the Ameba and cat images. It shows a very small numerical error. The average distance is less than 0.005. It means our proposed reconstruction algorithm is very accurate.

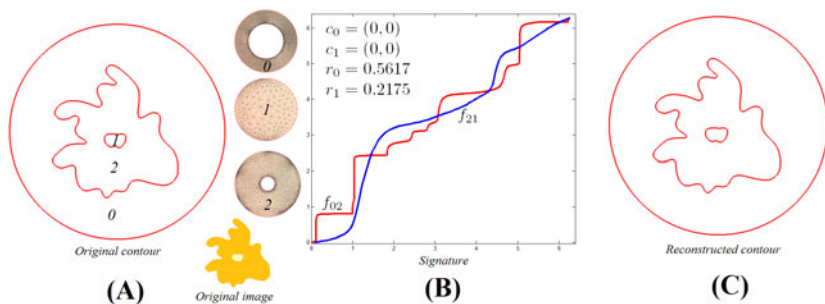
Table 2. Distance between the original and reconstructed contours

Ameba	Number of vertex	Distance sum	Average distance
Contour 1	685	1.669626	0.002437
Contour 2	112	0.238269	0.002127
Cat	Number of vertex	Distance sum	Average distance
Contour 1	96	0.227687	0.002372
Contour 2	92	0.295533	0.003212
Contour 3	363	1.674350	0.004613

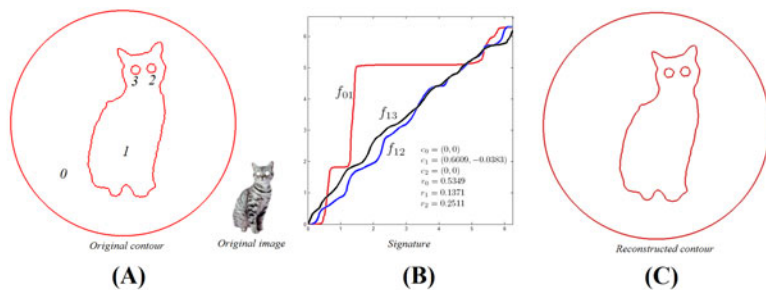


(a) Original contour (b) Reconstructed contour (c) Zoomed view (d) Zoomed view

**Fig. 8.** Comparison between the original contours (a) and the reconstructed ones (b). The zoomed views (c) and (d) show that the reconstructed ones are smoother



**Fig. 9.** Shape representation of the Ameba image and the reconstruction from its shape signature



**Fig. 10.** Shape representation of the cat image and its reconstruction from the shape signature

## 5 Conclusion and Future Work

We present a shape representation of multiply-connected planar domains using conformal geometry. Using conformal geometry, a set of diffeomorphisms from the unit circle  $\mathbb{S}^1$  to itself can be obtained, which together with the conformal modules are used to define the shape signature. We also introduce a reconstruction



algorithm to obtain shapes from their signatures. This completes the framework of our shape representation scheme. In the future, we will apply our algorithm for shape analysis based on Weil-Peterson metric. We will also test our proposed signatures on standard benchmark images, and compare with other existing representations for simply-connected shapes.

## Acknowledgement

This work is partially supported by NIH grant R01EB0075300A1, NSF IIS 0916286, CCF0916235, CCF0830550, III0713145, and ONR N000140910228.

## References

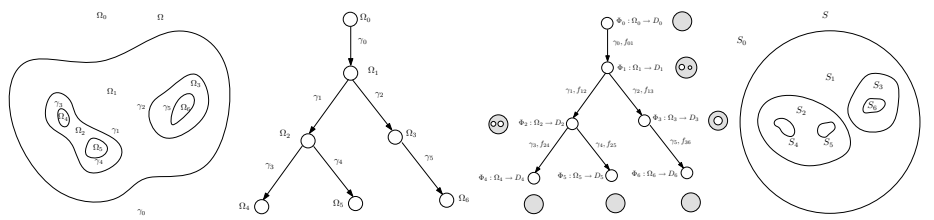
1. Zhu, S.C., Yuille, A.L.: A flexible object recognition and modeling system. *IJCV* 20, 8 (1996)
2. Liu, T., Geiger, D.: Approximate tree matching and shape similarity. In: *ICCV*, pp. 456–462 (1999)
3. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 24, 509–522 (2002)
4. Mokhtarian, F., Mackworth, A.: A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 14, 789–805 (1992)
5. Ericsson, A., Astrom, K.: An affine invariant deformable shape representation for general curves. In: *Proc. IEEE Intl. Conf. on Computer Vision*, vol. 2, pp. 1142–1149 (2003)
6. Sebastian, T., Klein, P., Kimia, B.: Shock based indexing into large shape databases. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*. LNCS, vol. 2352, pp. 731–746. Springer, Heidelberg (2002)
7. Dryden, I., Mardia, K.: *Statistical shape analysis*. John Wiley and Son, Chichester (1998)
8. Yang, Q., Ma, S.: Matching using schwarz integrals. *Pattern Recognition* 32, 1039–1047 (1999)
9. Lee, S.M., Clark, N.A., Araman, P.A.: A shape representation for planar curves by shape signature harmonic embedding. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, vol. 2, pp. 1940–1947 (2006)
10. Lipman, Y., Funkhouser, T.: Mobius Voting for Surface Correspondence. *ACM Transactions on Graphics (Proc. SIGGRAPH)* (August 2009)
11. Zeng, W., Zeng, Y., Wang, Y., Yin, X., Gu, X., Samaras, D.: 3D non-rigid surface matching and registration based on holomorphic differentials. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part III*. LNCS, vol. 5304, pp. 1–14. Springer, Heidelberg (2008)
12. Zeng, W., Lui, L.M., Gu, X., Yau, S.T.: Shape Analysis by Conformal Modules. *Methods Appl. Anal.* 15(4), 539–556 (2008)
13. Sharon, E., Mumford, D.: 2d-shape analysis using conformal mapping. *International Journal of Computer Vision* 70, 55–75 (2006)

14. Gardiner, F.P., Lakic, N.: Quasiconformal Teichmüller theory. American Mathematical Society, Providence (1999)
15. Henrici, P.: Applied and Computational Complex Analysis. Wiley Classics Library (1974)
16. Zeng, W., Yin, X.T., Zhang, M., Luo, F., Gu, X.: Generalized Koebe’s method for conformal mapping multiply connected domains. In: SPM 2009: 2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling, pp. 89–100 (2009)

### Appendix: Proof of Theorem 3

*Proof.* See Figure 11. In the left frame, a family of planar smooth curves  $\Gamma = \{\gamma_0, \dots, \gamma_5\}$  divide the plane to segments  $\{\Omega_0, \Omega_1, \dots, \Omega_6\}$ , where  $\Omega_0$  contains the  $\infty$  point. We represent the segments and the curves as a tree in the second frame, where each node represents a segment  $\Omega_k$ , each link represents a curve  $\gamma_k$ . If  $\Omega_j$  is included by  $\Omega_i$ , and  $\Omega_i$  and  $\Omega_j$  shares a curve  $\gamma_k$ , then the link  $\gamma_k$  in the tree connects  $\Omega_j$  to  $\Omega_i$ , denoted as  $\gamma_k : \Omega_i \rightarrow \Omega_j$ . In the third frame, each segment  $\Omega_k$  is mapped conformally to a circle domain  $D_k$  by  $\Phi_k$ . The signature for each closed curve  $\gamma_k$  is computed  $f_{ij} = \Phi_i \circ \Phi_j^{-1}|_{\gamma_k}$ , where  $\gamma_k : \Omega_i \rightarrow \Omega_j$  in the tree. In the last frame, we construct a Riemann sphere by gluing circle domains  $D_k$ ’s using  $f_{ij}$ ’s in the following way. The gluing process is of bottom up. We first glue the leaf nodes to their fathers. Let  $\gamma_k : D_i \rightarrow D_j$ ,  $D_j$  be a leaf of the tree. For each point  $z = re^{i\theta}$  in  $D_j$ , the *extension map*:  $G_{ij}(re^{i\theta}) = re^{f_{ij}(\theta)}$ .

We denote the image of  $D_j$  under  $G_{ij}$  as  $S_j$ . Then we glue  $S_j$  with  $D_i$ . By repeating this gluing procedure bottom up, we glue all leaves to their fathers. Then we prune all leaves from the tree. Then we glue all the leaves of the new tree, and prune again. By repeating this procedure, eventually, we get a tree with only the root node, then we get a Riemann sphere, denoted as  $S$ . Each circle domain  $D_k$  is mapped to a segment  $S_k$  in the last frame, by a sequence of extension maps. Suppose  $D_k$  is a circle domain, a path from the root  $D_0$  to  $D_k$  is  $\{i_0 = 0, i_1, i_2, \dots, i_n = k\}$ , then the map from  $G_k : D_k \rightarrow S_k$  is given by:  $G_k = G_{i_0 i_1} \circ G_{i_1 i_2} \circ \dots \circ G_{i_{n-1} i_n}$ . Note that,  $G_0$  is identity. Then the Beltrami coefficient of  $G_k^{-1} : S_k \rightarrow D_k$  can be directly computed, denoted as  $\mu_k : S_k \rightarrow \mathbb{C}$ . The composition  $\Phi_k \circ G_k^{-1} : S_k \rightarrow \Omega_k$  maps  $S_k$  to  $\Omega_k$ , because  $\Phi_k$  is conformal, therefore the Beltrami coefficient of  $\Phi_k \circ G_k^{-1}$  equals to  $\mu_k$ .



**Fig. 11.** Proof for the main theorem, the signature uniquely determines the family of closed curves unique up to a Möbius transformation

We want to find a map from the Riemann sphere  $S$  to the original Riemann sphere  $\Omega$ ,  $\Phi : S \rightarrow \Omega$ . The Beltrami-coefficient  $\mu : S \rightarrow \mathbb{C}$  is the union of  $\mu_k$ 's each segments:  $\mu(z) = \mu_k(z), \forall z \in S_k$ . The solution exists and is unique up to a Möbius transformation according to Quasi-conformal Mapping theorem [14].

Note that, the discrete computational method is more direct without explicitly solving the Beltrami equation. From the Beltrami coefficient  $\mu$ , one can deform the conformal structure of  $S_k$  to that of  $\Omega_k$ , under the conformal structures of  $\Omega_k$ ,  $\Phi : S \rightarrow \Omega$  becomes a conformal mapping. The conformal structure of  $\Omega_k$  is equivalent to that of  $D_k$ , therefore, one can use the conformal structure of  $D_k$  directly. In discrete case, the conformal structure is represented as the angle structure. Therefore in our algorithm, we copy the angle structures of  $D_k$ 's to  $S$ , and compute the conformal map  $\Phi$  directly.

# A Coarse-to-Fine Taxonomy of Constellations for Fast Multi-class Object Detection<sup>\*</sup>

Sanja Fidler<sup>1,2</sup>, Marko Boben<sup>1</sup>, and Aleš Leonardis<sup>1</sup>

<sup>1</sup> University of Ljubljana

<sup>2</sup> UC Berkeley EECS and ICSI

fidler@eecs.berkeley.edu, {marko.boben, ales1}@fri.uni-lj.si

**Abstract.** In order for recognition systems to scale to a larger number of object categories building visual class taxonomies is important to achieve running times logarithmic in the number of classes [12]. In this paper we propose a novel approach for speeding up recognition times of multi-class part-based object representations. The main idea is to construct a taxonomy of constellation models cascaded from coarse-to-fine resolution and use it in recognition with an efficient search strategy. The taxonomy is built *automatically* in a way to minimize the number of expected computations during recognition by optimizing the cost-to-power ratio [3]. The structure and the depth of the taxonomy is not pre-determined but is inferred from the data. The approach is utilized on the hierarchy-of-parts model [4] achieving efficiency in both, the representation of the structure of objects as well as in the number of modeled object classes. We achieve speed-up even for a small number of object classes on the ETHZ and TUD dataset. On a larger scale, our approach achieves detection time that is logarithmic in the number of classes.

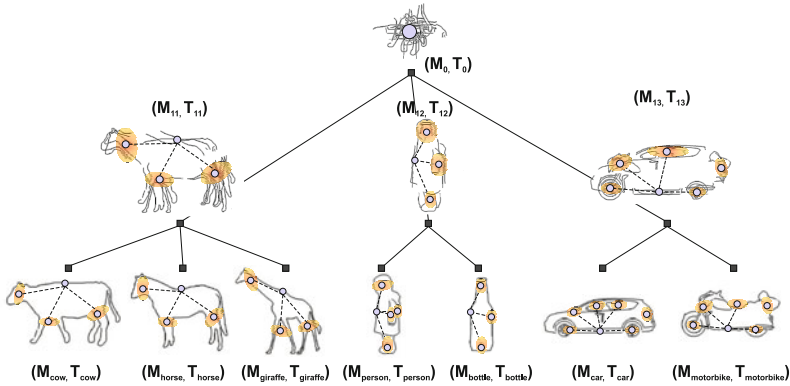
## 1 Introduction

Representing objects as spatial layouts of simpler parts has been shown as an effective way of modeling generic object classes [5,6,7]. In order to recognize and detect a larger number of object categories in images, several works have proposed feature sharing among the objects to achieve better generalization as well as to cut down computation time [5,7]. However, these approaches still run with time linear in the number of classes [8], since they need to scan the (shared) feature space with a detector for each class separately. In contrast, visual class taxonomies [2,9] induce a hierarchy over the class labels: usually a hierarchical tree of classifiers is used to achieve recognition complexity logarithmic in the number of classes [1].

In this paper we propose a novel approach for speeding up multi-class object detection based on *part-based object representations* [10,7] by constructing a

---

<sup>\*</sup> This research has been supported in part by the following funds: EU FP7-215843 project POETICON, and Slovenian Research Agency (ARRS) research program Computer Vision P2-0214, and ARRS research projects J2-3607 and J2-2221.



**Fig. 1.** A hierarchical tree of coarse-to-fine constellation models.  $M$ 's denote the models,  $T$ 's denote the tests (detectors for the models), explained in Sec. 4.1 and 5. The leaf nodes (object class models) are assumed known (we use 4 to learn them), whereas all the other models in the taxonomy are obtained automatically, by clustering the object models in the leaves. The depth and structure of the taxonomy are learned.

visual taxonomy of constellation models cascaded from coarse-to-fine resolution (Fig. 1). The taxonomy is generative, where each object class can be generated by following a particular path in the hierarchical taxonomy tree. The taxonomy is constructed by clustering a set of object class models into a hierarchical tree of increasingly coarser constellation models both in terms of structure as well as in the appearance of parts. The tree is built *automatically* in a way to minimize the number of expected computations during recognition by optimizing the cost-to-power ratio 3. The structure and the depth of the taxonomic tree are not pre-determined but are inferred from the data. During recognition, our approach uses the learned taxonomy to *prune* the search space in a coarse-to-fine fashion, starting from the root using the depth-first search algorithm.

The approach is utilized on the hierarchy-of-parts model 4 achieving efficiency in both, the representation of the shape of the objects (by using a hierarchy of shareable shape features that gradually progress in complexity) as well as in the number of modeled object classes (by inducing a taxonomy over the class labels). Compared to the baseline 4, we demonstrate good speed-up even for a small number of object classes on the ETHZ 11 and TUD dataset 12. On a larger scale (Caltech 101 13 and LabelMe 5), our approach achieves detection time that is logarithmic in the number of classes.

## 2 Related Work

Prior work on multi-class object recognition is mainly concerned with speeding-up *classification* approaches 1,8,14, while our goal here is to speed-up *generative*, part-based object class models. Nevertheless, the ideas behind these approaches are related to ours. Zhender et al. 14 employ a hierarchical cascade of classifiers, while 8 translates the classification stage into matching in

a high-dimensional vector space, where fast, scalable solutions exist. Similarly, Stewenius and Nister [15] build a hierarchical cluster tree in the vector space (representing descriptor appearance) in an image retrieval scenario. Our problem is substantially different since we are dealing also with object geometry.

Bart et al. [9] build a taxonomy of images represented as bags of visual words in an unsupervised manner by learning a hierarchical tree of topic models. Their taxonomy is generative like ours, however, they are dealing with taxonomy of images and not objects and the representation does not take into account the geometry between features. Closer to our approach is the work by Sivic et al. [2], where the authors build a taxonomy of increasingly coarser object representations both in terms of the spatial layout (fixed grid with varying degree of resolution) and appearance (by varying the degree of clustering of SIFT features) using a Hierarchical LDA model. In our approach we use the constellation model [16,10,4] as the means to represent the objects and show how to build a generative taxonomy of increasingly coarser constellations.

Our work is also related to coarse-to-fine model matching which has appeared in many forms in the literature. Gavrilu [17] proposed a method for hierarchical clustering of object templates and applying the cascade to speed-up template matching in the domain of pedestrian detection.

In [18], the authors propose a cascade of detectors based on the constellation model to detect *specific instances* of objects. Similarly as in our approach, the detectors are cascaded in a coarse-to-fine resolution, however, the coarsening of the spatial grid is pre-defined, while in our approach the coarsening of both the location and appearance is *learned* by optimizing the cost-to-power ratio on a set of training images. We also deal with generic object classes.

Amit and Geman [19] learn “spread” tests to check multiple object hypotheses simultaneously. Spreading corresponds to OR-ing (disjunctions) the locations of simple oriented edges so that the conjunction of these coarsely positioned features is common (shared) among several object classes. In our approach the spreading (OR-ing) is done in both location as well as in appearance making the method generally applicable to part-based models.

Our work builds on some of the theoretical ideas on coarse-to-fine search strategies [3,20]. We apply them to our particular problem, which is building visual taxonomies of object classes for part-based object representations.

Note that our constellation-based object taxonomy differs from hierarchical representations of object structure [4,21]. Here we induce a hierarchy on the *object class labels*, while [4,21] deal with a hierarchy of *shape features*.

### 3 Overview and Contributions

The problem we are tackling is multiple object class detection using part-based object representations, in particular the constellation-type models [16,10]. We assume that we have available constellation models for a set of object classes and we want to perform recognition with them in arbitrary images (we do not know which objects are present in an image). In the original approach [16,10], a

detector for each class separately needs to be run on a query image resulting in recognition times linear in the number of modeled objects.

The novel idea of this paper is to speed-up recognition by using a generative taxonomy of constellation object detectors organized hierarchically in a coarse-to-fine manner. Recognition will proceed from the root down by first employing a small set of coarse detectors and pruning improbable search paths. Performed in this way, the detectors in the leaves, which are the given object class models, will rarely be implemented, thus speeding up the overall recognition procedure.

This paper makes three novel contributions:

1. **Representing a visual taxonomy of object classes** with a coarse-to-fine taxonomy of constellation models.
2. **Automatic construction of the taxonomy** by minimizing the expected number of computations during recognition. We will build on the coarse-to-fine search strategies proposed by Blanchard and Geman [3].
3. **Combining the class taxonomy with a structure hierarchy** for fast multi-class object recognition.

The approach is organized as follows. In Sec. 4 we present the representation of the constellation taxonomy which will be referred to as the *taxonomic constellation tree (TCT)*. Sec. 5 explains the recognition procedure using TCT. In Sec. 6 we propose an approach to automatic construction of the TCT model.

## 4 Representation: Coarse-to-Fine Constellation Taxonomy

Let  $\mathcal{C}$  be a set of classes. We assume we have available a *constellation-type model* [10]  $M_c$  for each class  $c \in \mathcal{C}$ , the collection of which forms our *database*  $\mathcal{M} = \{M_c\}_{c \in \mathcal{C}}$  of object models. Note that each object class can be represented with (a mixture of) multiple models (e.g. a model per view or object articulation), however for the ease of exposition, we assume the existence of one model per class. We first define the basic constellation model in Subsec. 4.1. In Subsec. 4.2 we propose the representation using a taxonomic constellation tree.

### 4.1 The Probabilistic Constellation Model of Objects

From a query image  $I$  a set of features  $\mathbf{F}$  along with their locations  $\mathbf{X}$  is first extracted. We assume that features are discrete, i.e., each feature in  $\mathbf{F}$  is characterized by *type* (e.g. a particular shape or visual word). The set of all feature types forms a *vocabulary*. We assume the object models to have a star topology, although the approach can be easily extended to more complex topologies.

Each model represents an object class as a collection of parts and spatial relations among them. We follow [10] to define the model. Each class  $c \in \mathcal{C}$  is represented with a model  $M_c = (P_c, \theta_c^{app}, \theta_c^g)$ ,  $M_c \in \mathcal{M}$ , which has  $P_c$  parts, appearance parameters  $\theta_c^{app}$  for the parts, and geometry parameters  $\theta_c^g$ , where the positions of all parts are conditioned on the position of a so-called *reference*

part, but are mutually independent conditionally on the reference part. We additionally define the assignment variable  $\mathbf{h}$  of size  $P_c$ , which assigns some features in  $\mathbf{F}$  to the parts in the model  $M_c$ .

The joint density is factored as follows [10]:

$$p(\mathbf{F}, \mathbf{X}, \mathbf{h} | c, M_c) = \underbrace{p(\mathbf{F} | \mathbf{h}, c, M_c)}_{\text{appearance}} \underbrace{p(\mathbf{X} | \mathbf{h}, c, M_c)}_{\text{geometry}} \underbrace{p(\mathbf{h} | c, M_c)}_{\text{occlusion}}$$

For the appearance term we have the following factorization:

$$p(\mathbf{F} | \mathbf{h}, c, M_c) = \prod_{i=1}^{P_c} p(\mathbf{F}(h_i) | c, \theta_{c_i}^{app}) \quad (1)$$

We will assume that each part corresponds to one feature type or is modeled as a distribution over a small number of feature types, i.e.  $p(f | c, \theta_{c_i}^{app}) > 0$  for a small number of all types  $f$  in the vocabulary.

The geometry term can be factorized as follows:

$$p(\mathbf{X} | \mathbf{h}, c, M_c) = p(\mathbf{x}_R | h_R) \prod_{j \neq R} p(\mathbf{x}_j | \mathbf{x}_R, h_j, c, \theta_{c_j}^g),$$

where  $R$  denotes the reference part. The distribution  $p_{jR} := p(\mathbf{x}_j | \mathbf{x}_R, h_j, c, \theta_{c_j})$  is taken to be Gaussian [10, 4], i.e.,  $p_{jR} = \mathcal{N}(x_j - x_R | \theta_{c_j})$ , where  $\theta_{c_j} = (\mu_{c_j}, \Sigma_{c_j})$ .

In recognition, a decision whether an object of class  $c$  is present in an image (at a particular location) or not is made on the likelihood-ratio [16]:

$$\frac{p(c | \mathbf{F}, \mathbf{X})}{p(B | \mathbf{F}, \mathbf{X})} \propto \frac{\sum_{\mathbf{h}} p(\mathbf{F}, \mathbf{X}, \mathbf{h} | c, M_c)}{p(\mathbf{F}, \mathbf{X}, \mathbf{h}_0 | B)}, \quad (2)$$

where  $B$  denotes the background (“no object of class  $c$  present”) and  $\mathbf{h}_0$  is the null hypothesis explaining all features as background. The occlusion term can be defined as in [10], however, we will not explicitly deal with it in this paper.

## 4.2 A Coarse-to-Fine Taxonomic Constellation Tree (TCT)

Our approach exploits the fact that constellation models for similar object classes have (or share) similar spatial arrangements of parts (of possibly similar appearance) and can thus be grouped in a natural way: we can design “coarse” constellations that capture the distribution over *multiple* object classes. In particular, we will cluster the models in the database  $\mathcal{M}$  to obtain a hierarchical tree  $\mathcal{H}$  of constellations, where the constellations close to the root are coarse in both geometry and appearance and span the distribution over a larger number of object classes, while the constellations closer to the leaves are more specific. The leaves of the tree contain the given object constellation models from  $\mathcal{M}$ . This induces a taxonomic organization on object classes which will be used during detection to speed-up the computations. Fig. 1 illustrates  $\mathcal{H}$ .



A *taxonomic constellation tree* (TCT)  $\mathcal{H}$  is represented with a tree graph,  $\mathcal{H} = (V, E)$ , where each node  $\xi \in V$  in the tree contains a model  $M_\xi = (P_\xi, \theta_\xi^{app}, \theta_\xi^g)$  and a corresponding test  $T_\xi$  (which will be explained in more detail in Sec. 5). The leaves of the tree correspond to the object models  $M_c \in \mathcal{M}$  and are assumed to be known in advance. All other models in the tree are obtained by hierarchical clustering of the models in  $\mathcal{M}$ . By definition, the root node will be set at level 0 of the tree.

Define the *domain*  $D(M_\xi)$  of a model  $M_\xi$  to be the set of object classes  $\{c\}$ , such that the likelihood  $p(I(c)|M_\xi)$  of observing an image  $I(c)$  of any of these classes under the model  $M_\xi$  is non-zero, or rather, higher than a threshold  $\tau_\xi$  (discussed in Sec. 5):  $D(M_\xi) = \{c \in \mathcal{C} : p(I(c)|M_\xi) > \tau_\xi\}$ . A node  $\xi$  from level  $\ell$  is a parent of node  $\eta$  from level  $\ell + 1$  below, if the domain of the model  $M_\eta$  is entirely contained in the domain of  $M_\xi$ :  $D(M_\eta) \subset D(M_\xi)$ . This means that if the likelihood of class  $c$  is high under  $M_\eta$  it must also be sufficiently high under its parent  $M_\xi$ . Thus a model that has multiple children spans a distribution over *multiple* object classes, and the variances of the distribution in both appearance of parts as well as their geometry have to be large enough to accommodate for this. In the root node, we put a “trivial” model  $M_0$  that has only one part, where  $\theta_0^{app}$  is uniform over the feature space and  $\theta_0^g$  is trivial.

## 5 Efficient Inference

The main purpose of TCT is to reduce the number of full object class models that need to be evaluated at a particular region in an image during recognition. To achieve this we will evaluate TCT from the root down in a depth-first search manner and *prune* the improbable search paths (the unlikely object hypotheses).

For the purpose of pruning, we will construct a binary computationally inexpensive *test*  $T_\xi$  for each model  $M_\xi$  in  $\mathcal{H}$ . The test will be, during recognition, used to form a decision whether the children of the model  $M_\xi$  should be further explored or not. A test will yield a value 1 if the finer hypotheses (the children of  $\xi$ ) are “worth” evaluating further, and 0 if the entire subbranch of hypotheses can be reliably eliminated from the search process. During training, each test will be designed to have no missed detections, at the expense of having some false positives. This idea is adopted from [19]. The false positives during recognition mean that some search paths will get explored even though the corresponding object classes might not be present. If some search path ends in a leaf (meaning that no parents yielded a negative answer), the likelihood-ratio as defined in (2) will be evaluated for the corresponding object class model and a full probabilistic decision will be made on the presence or absence of an object.

A test  $T_\xi(I)$  will take the following form:

$$\begin{aligned} T_\xi(I) &= \mathbf{1}(J_\xi(I) > \tau_\xi), \quad \text{where} \\ J_\xi(I) &= \max_{\mathbf{h}} p(\mathbf{F}, \mathbf{X}, \mathbf{h} \mid \xi, \theta_\xi) \end{aligned} \quad (3)$$

where  $\mathbf{1}(\cdot)$  is the indicator function. It checks whether the most probable hypothesis under the model  $M_\xi$  is higher than a particular threshold  $\tau_\xi$ . We use

the *max* instead of a *sum* (which would correspond to the image likelihood under the model  $M_\xi$ ) since it allows for further speed-ups as discussed below. If the probability of the most likely hypothesis is too small the test will be 0 and none of the children of  $M_\xi$  in the TCT will be evaluated further. If  $T_\xi(I)$  is 1 the children of  $M_\xi$  get tested in a similar way. We adopt a depth-first search as in [3,20], meaning that the child  $\mu$  with the highest value of  $J_\mu$  is explored first.

By using the tests as defined in (3), we assume the existence of thresholds  $\tau_\xi$  which effectively separate the “foreground” (classes in the domain  $D(M_\xi)$ ) from the “background”, similarly as in [19]. As in [19], the thresholds  $\tau_\xi$  will be learned such that the probability of a missed detection will be 0 (or close to 0):  $p(T_\xi = 0|D(M_\xi)) = 0$ , while the number of false positives will be sufficiently small:  $p(T_\xi = 1|background, M_\xi) \ll 1$ .

For further speed-up, we additionally define the individual likelihoods of the appearances and geometry under  $M_\xi$ :  $J_{\xi,j}^{app}(I) = \max_{h_j} p(\mathbf{F}(h_j)|M_\xi)$  and  $J_{\xi,j}^g(I) = \max_{h_j} p(\mathbf{x}_j|\mathbf{x}_R, h_j, M_\xi)$ . While evaluating the tests in an image we can exploit the factorization of  $p(\mathbf{F}, \mathbf{X}, \mathbf{h} | \xi, \theta_\xi)$ : since the overall probability  $J_\xi(I)$  should exceed the threshold  $\tau_\xi$ , each of the individual probabilities  $J_{\xi,j}^{app}(I)$  and  $J_{\xi,j}^g(I)$  in the product should also satisfy this condition. This limits the allowed geometry to a small region making computations fast, while at the same time quickly prunes off assignments where the individual likelihoods are small.

## 6 Learning the Taxonomic Constellation Tree

Given the database of object class models  $\mathcal{M}$ , our goal is to construct a taxonomic constellation tree  $\mathcal{H}$  by optimizing its expected run time on a set of training images. We adopt some of the conceptual ideas from [20,3], i.e., optimizing the power-to-cost ratio to learn the representation.

We first introduce the notation. Let  $\mathcal{H}^\ell = (V^\ell, E^\ell)$  denote the subgraph of  $\mathcal{H}$  at level  $\ell$  with the set of models  $\mathcal{M}^\ell$  and tests  $T^\ell$  defined in the nodes  $V^\ell$ . Denote with  $\theta^{g,\ell}$  and  $\theta^{app,\ell}$  the parameters of the models at level  $\ell$  and with  $\mathcal{V}^\ell$  the vocabulary of feature types at this level. Further, let  $B(M_\xi)$  denote the background interpretation under  $M_\xi$ , which is any interpretation outside of the model’s domain  $D(M_\xi)$ . Define the missed detection rate of a test with  $\alpha(T_\xi) = p(T_\xi = 0|D(M_\xi))$ . Each test will be assigned a *cost* denoted with  $t(T_\xi)$ , which can be measured with dedicated CPU time, and *power* denoted with  $\beta(T_\xi) = p(T_\xi = 0|B(M_\xi))$ . The power is thus the selectivity of the test. It is shown in [3], that under certain conditions, a sufficient condition for the optimality of the coarse-to-fine search is:

$$\forall \xi \in V : \frac{t(T_\xi)}{\beta(T_\xi)} \leq \sum_{\eta=\text{child}(\xi)} \frac{t(T_\eta)}{\beta(T_\eta)}, \quad (4)$$

which will motivate our learning objective function. The tree will be built from the leaves up. The models at a particular level  $\ell$  will be obtained by clustering the models from the previous level,  $\ell + 1$  (the root node will be at level 0). Each

(clustered) model  $M_\xi \in \mathcal{H}^\ell$  will be evaluated with respect to the effectiveness of its corresponding test (detector)  $T_\xi$ .

Assume we already have a model  $M_\xi$ . Its test  $T_\xi$  (which has one parameter: a threshold  $\tau_\xi$ ) is learned as follows. Since we want the missed detection rate to be close to zero,  $\alpha(T_\xi) \approx 0$ , we set  $\tau_\xi$  as the minimum of  $\{J_\xi(I)\}_{I \in \text{train}}$ , or rather, some percentage of it to allow for generalization outside the training data [19][22]. When the test is learned, we can calculate its cost  $t$  and the power  $\beta$  from the training data. For better generalization, we use different training data to learn the tests than the data used to train the original set of models  $\mathcal{M}$ .

Our learning objective for each level  $\ell$  is to optimize the cost-to-power ratio:

$$\mathcal{H}_*^\ell = \arg \min_{\mathcal{H}^\ell} \sum_{\xi \in V^\ell} \frac{t(T_\xi)}{\beta(T_\xi)}, \tag{5}$$

where each  $T_\xi$  also satisfies the condition in (4). Finding the global maximum of (5) is obviously intractable, thus our solution will be to generate good guesses and use stochastic optimization to improve the result. In particular, our learning approach uses the following steps: 1.) **parameter clustering**: cluster the parameters  $\theta^g$  and the vocabulary of feature types  $\mathcal{V}$  from the models at the level below, 2.) **model clustering**: cluster the models from the level below with respect to the appearance and geometry clusters, 3.) **local optimization**: optimize  $\mathcal{H}^\ell$  on the training data. Steps 1 to 3 are repeated several times (by varying the clustering parameters) and the  $\mathcal{H}^\ell$  with the lowest cost in (5) is selected.

**Parameter clustering.** To cluster the models, geometry will be the primary cue: models with similar spatial arrangements of parts will have a higher chance of being merged. We start by clustering the geometry parameters and group the feature types afterwards. *Geometry*: We use the k-means algorithm to cluster the means  $\mu^{\ell+1}$  of the Gaussian parameters  $\theta^{g,\ell+1}$  (defined in Sec. 4.1) and assign each mean  $\mu^{\ell+1}$  to its closest cluster. *Appearance*: With respect to the centers, we can calculate the similarity between different types of features from vocabulary  $\mathcal{V}^{\ell+1}$  and use a clustering algorithm to group them, which yields the vocabulary  $\mathcal{V}^\ell$ . Clustering in the case of discrete feature types means OR-ing (disjunction). To calculate the similarity we can do the following: for every pair of models in which a part of one model coincides with a part of the other (the means of the geometric distributions fall in the same k-means cluster) we simply increase the similarity score of the feature types corresponding to these two parts. With respect to  $\mathcal{V}^\ell$  we can re-define the appearance parameters  $\theta^{app,\ell+1}$ .

**Model clustering.** If we do not consider the variances of the geometric distributions, we can now group the models  $M^{\ell+1}$  which have the same geometric means (k-means clusters) as well as the appearance distributions  $\theta^{app,\ell+1}$  into one model  $M^\ell$  at level  $\ell$  and make corresponding edges in the graph  $\mathcal{H}^\ell$  (to models that were merged into  $M^\ell$ ). For each  $M^\ell$  we can estimate the variances of the geometric distributions from the variances of its children at level  $\ell + 1$ .

**Local optimization.** Since the model clustering stage disregards the information on the variances of the geometry, the resulting models can have too large

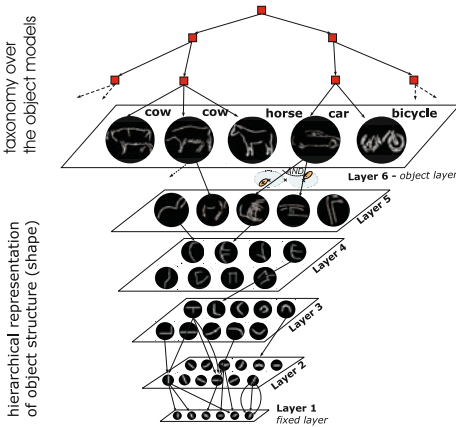
variances and the corresponding tests will not be very selective. This means that the condition in (4) will not be fulfilled. It is thus necessary to optimize the clusters: we can remove one edge from  $\mathcal{H}^\ell$ , re-estimate the variances of the parent model  $M_\xi^\ell$  and re-calculate the cost-to-power ratio  $t(T_\xi^\ell)/\beta(T_\xi^\ell)$ . At each step the edge with the highest cost-to-power ratio is removed from  $\mathcal{H}^\ell$ .

The levels of TCT are built until there is still a computational saving with respect to the cost-to-power ratio (5).

## 7 Combining TCT with the Hierarchy-of-Parts Model (4)

To represent, learn and detect object classes we will use the hierarchy-of-parts model (4), which will serve as the *baseline* in the experiments. We give a brief summary of the model. Objects are represented with a recursive compositional shape vocabulary, the structure of which is learned from images without supervision. The vocabulary contains a set of shape *compositions* at each layer. Each composition is defined recursively: it is a hierarchical generative probabilistic model that represents a geometric configuration of a small number of parts which are themselves hierarchical probabilistic models, i.e., compositions from a previous layer of the vocabulary. Different compositions can share models for the parts, which makes the vocabulary efficient in size and results in faster inference.

The definition of a composition with respect to just one layer below is akin to that of the constellation model (10). Each part is spatially constrained on the parent composition via a spatial relation which is modeled with a two-dimensional Gaussian distribution. Each part in a composition has also an “appearance” which is defined as a (discrete) distribution over the set of compositions from the previous layer of the vocabulary. At the lowest layer, the vocabulary consists of a small number of short oriented contour fragments. The vocabulary at the top-most layer contains compositions that represent the shapes of the whole objects. Altogether six layers are learned.



**Fig. 2.** Hierarchy-of-parts model (4) (hierarchy of shape features of increasing complexity) combined with the proposed taxonomic constellation tree TCT (taxonomy of object class models). The structure hierarchy has six layers. The taxonomy is built over the last layer’s models which are the compositions (constellations) modeling the whole shape of the objects. The depth of the taxonomy is not pre-determined.

We will combine our taxonomy model with the hierarchical vocabulary in the following way. The top layer in the vocabulary is an object layer and grows linearly with the number of modeled classes. Thus the models at the final layer form our database of models for which the taxonomy is built over. The features  $\mathbf{F}$  which are extracted from an image prior to recognition will be the detections corresponding to the compositions in the fifth layer of the vocabulary (one layer below the object layer). The illustration of the combined model is given in Fig. 2.

## 8 Experimental Results

The aim of the experiments is to compare the speed of the proposed taxonomy model vs the baseline as the number of classes increases, and show that its detection rate does not degrade significantly with respect to the baseline model.

### 8.1 ETHZ Shape Dataset: 5 Object Classes

We use the ETHZ shape dataset of 5 object classes: apple logo, bottle, giraffe, mug and swan. For training and testing we follow the same protocol as in [11]. For this and all experiments in the paper, the following was used to train the model. Bounding boxes (scaled to size approx. 200 pixels in diagonal) were used in training. Two image scales, spaced apart by  $\sqrt{2}$ , were used to train object models, while 4 to 6 scales were used in testing. For constructing the taxonomy the positive training images as well as a set of 100 natural images not containing the objects were used to construct TCT. The positive training images available in each dataset were split into 5 images used as positive examples for test learning and the rest for learning the object representation, i.e., training the hierarchy-of-parts model [4].

For the five classes, the learned TCT consisted of 2 layers. The detection times are reported in Fig. 3 (the first five classes in the Shape-15 plot). The times reported do not include the process of feature extraction. Since both the baseline and our approach involve the same feature extraction process, such a comparison gives a clearer picture of the achieved speed-up.

The detection performance is reported in Table 1. The TCT approach performs slightly worse than the baseline which is due to the depth-first search strategy during recognition: while it boosts recognitions times, the performance might slightly degrade.

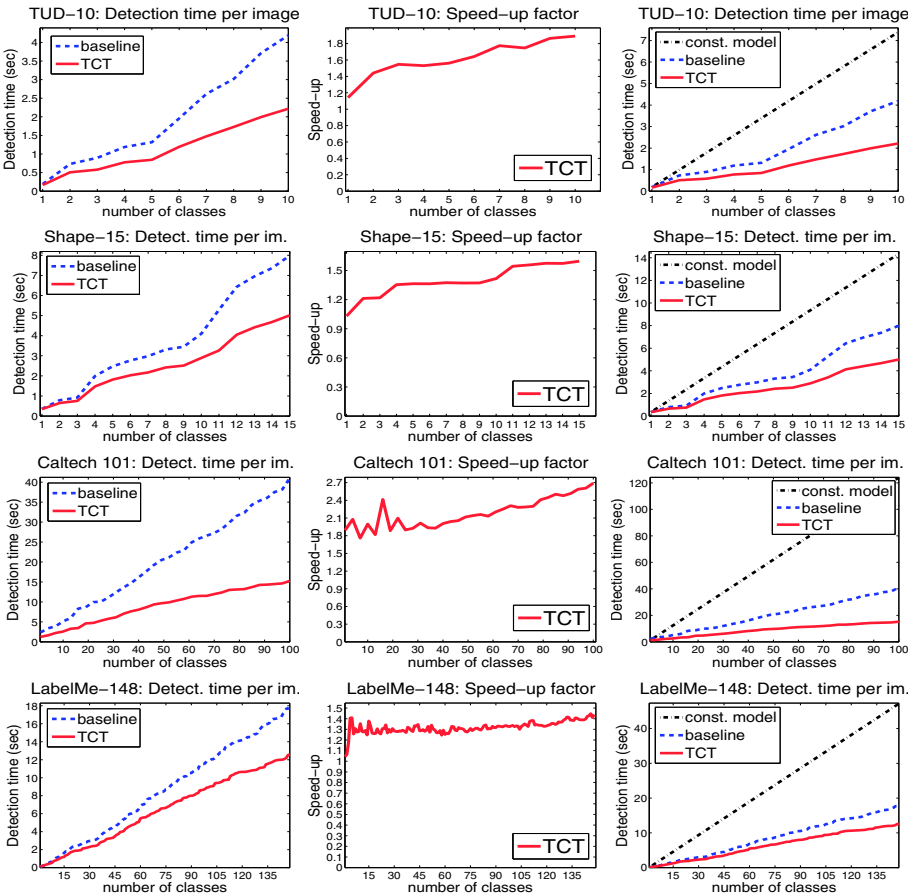
**Table 1.** Average detection-rate (in %) at 0.4 FPPI for the ETHZ dataset [11]

	applelogo	bottle	giraffe	mug	swan	average
[11]	83.2(1.7)	83.2(7.5)	58.6(14.6)	83.6(8.6)	75.4(13.4)	76.8
[23]	95.0	89.3	75.4	90.3	94.1	88.8
baseline	88.2(3.4)	87.6(1.5)	83.5(1.1)	86.1(2.)	80(3.5)	85.1
TCT	87.3(2.6)	87.3(2.2)	83.1(1.3)	85.8(3.1)	79.4(5.4)	84.6

### 8.2 TUD Shape Dataset: 10 Object Classes

TUD *shape2* [12] contains 10 classes of home appliances such as knife, fork, mug, saucepan, etc. The test set contains 10 images of objects for each class, and roughly 100 for training – from these we randomly choose 20 images for training the object class models with [4] and another 5 for training the TCT (with the addition of 100 negative natural images). The learned TCT consisted of 2 layers.

The recognition times of both the baseline and TCT are plotted in Fig. 3. The speed-up factor for 10 classes is almost 2 and grows as more classes are added. We also compare the baseline (hierarchy-of-parts [4]) and TCT against the recognition procedure used in the original constellation model [16,10]. There, a detector for each class is run separately over the image, thus resulting in



**Fig. 3.** Comparison between baseline and proposed TCT model for several datasets in: **a)** average detection times per image, **b)** speed-up of TCT over baseline, **c)** comparison of detection times of baseline and TCT against the recognition procedure of the constellation model [16] (see text for details), all as a function of the number of classes

complexity linear in the number of classes, i.e.,  $n \cdot O(\text{detector for one class})$  (where  $n$  is the number of classes). Due to part sharing and the indexing and matching recognition scheme in [4], not all class models get evaluated in each image location already for our baseline, however, the running time of [4] is still linear, but with a smaller constant, i.e.,  $k \cdot O(\text{detector for one class})$  (where  $k \ll n$ ).

The classification accuracy of the baseline approach is 69%, while the TCT achieves a 66% classification rate. For comparison, Stark et al. [12] report a 44% accuracy using a discriminative approach (SVM over different types of features).

### 8.3 Shape 15

The approach was also tested on detection of 15 shape-based object classes. The first 5 are taken from ETHZ [11] and the 10 additional from the GRAZ dataset [7]. The comparison in detection performance is given in Fig. 4, while the comparison in detection times is plotted in Fig. 3. TCT is depicted in Fig. 6.

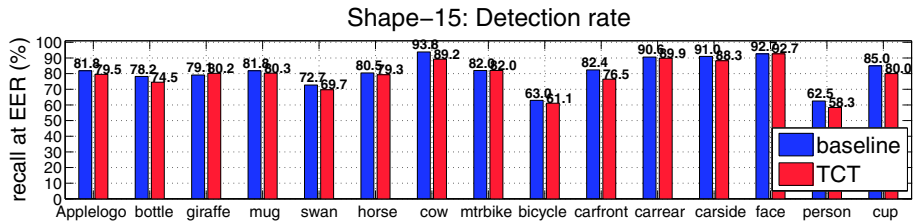


Fig. 4. Comparing detection rates (recall at EER) on Shape-15 – composed of the ETHZ [11] (first 5 classes in the plot) and 10 classes from GRAZ dataset [7]

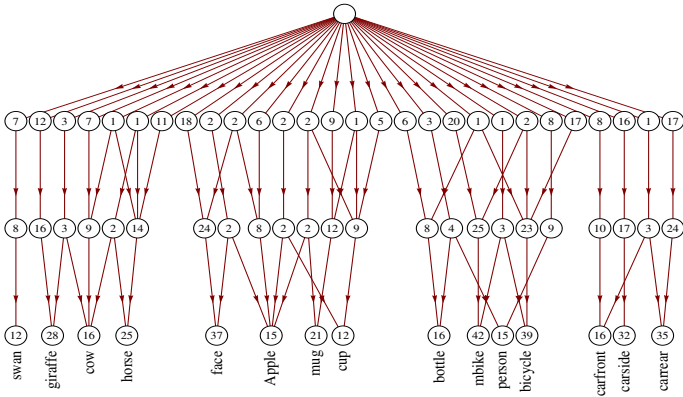


Fig. 5. Detection examples: *horse, motorbike, giraffe, cow, bottle*

### 8.4 Caltech 101 and LabelMe Dataset (148 Object Classes)

The approach has been also tested on a larger scale. We used 100 classes from the Caltech-101 dataset [13] and 148 classes from the LabelMe dataset [24]. For both datasets, 30 images were randomly chosen from the available annotations to train each class. For both datasets the taxonomy resulted in a 4 layer TCT. The approach was tested for detection time on 200 images randomly sampled from each dataset. Evaluation is done on full images, i.e., of approx. size  $500 \times 400$ . The comparison in detection times (averaged over 200 images) is plotted in Fig. 3.

The speed-up on the LabelMe dataset is not very high due to various possible reasons: 1.) too few classes are still being used to adequately showcase a taxonomy (as also reported in [8]), 2.) the LabelMe annotations are noisy and thus



**Fig. 6.** TCT learned on Shape-15. Bottom level are object classes – the number denotes the number of models for each class.

the baseline does not learn the classes very well, 3.) the images used for testing in LabelMe are images of complex scenes containing many objects and significant texture (with which already the baseline [4] does not deal well) resulting in many false-positives decisions, 4.) a number of classes have very simple shape (e.g. a *street lamp* which mostly gets represented as a long vertical line) and thus lead to a higher number of false positives per image already with the baseline method. A detection (even if it is a false positive) means that a whole branch in the TCT tree needs to be explored as well, making the speed-up smaller.

## 9 Summary and Conclusions

In this paper we proposed a novel approach for automatic construction of a generative visual class taxonomy with the aim to speed-up multi-class object detection with the constellation models. Specifically, we learned a taxonomic tree of constellations cascaded from coarse-to-fine resolution, and the corresponding detectors which take the form of inexpensive, binary tests. Both the taxonomic constellation tree and the corresponding tests are *learned* from the training data.

The approach has been utilized on the hierarchy-of-parts model [4] achieving efficiency in both, the representation of the object structure as well as in the number of modeled object classes. We demonstrated good speed-up on several recognition datasets with promising scaling tendency for recognition on a larger scale. As part of future work, we plan to add discriminative information to the taxonomy to boost also its recognition accuracy.

## References

1. Marszałek, M., Schmid, C.: Constructing category hierarchies for visual recognition. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 479–491. Springer, Heidelberg (2008)



2. Sivic, J., Russell, B.C., Zisserman, A., Freeman, W.T., Efros, A.A.: Unsupervised discovery of visual object class hierarchies. In: CVPR (2008)
3. Blanchard, G., Geman, D.: Hierarchical testing designs for pattern recognition. *Annals of Statistics* 33, 1155–1202 (2005)
4. Fidler, S., Leonardis, A.: Towards scalable representations of visual categories: Learning a hierarchy of parts. In: CVPR (2007)
5. Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing visual features for multiclass and multiview object detection. *IEEE PAMI* 29, 854–869 (2007)
6. Huttenlocher, D., Felzenszwalb, P.: Pictorial structures for object recognition. *IJCV* 61, 55–79 (2005)
7. Opelt, A., Pinz, A., Zisserman, A.: Learning an alphabet of shape and appearance for multi-class object detection. *IJCV* 80, 16–44 (2008)
8. Stefan, A., Athitsos, V., Yuan, Q., Sclaroff, S.: Reducing jointboost-based multi-class classification to proximity search. In: CVPR (2009)
9. Bart, I., Porteous, E., Perona, P., Wellings, M.: Unsupervised learning of visual taxonomies. In: CVPR (2008)
10. Fergus, R., Perona, P., Zisserman, A.: Weakly supervised scale-invariant learning of models for visual recognition. *IJCV* 71, 273–303 (2007)
11. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Accurate object detection with deformable shape models learnt from images. In: CVPR (2007)
12. Stark, M., Schiele, B.: How good are local features for classes of geometric objects?. In: ICCV (2007)
13. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: IEEE CVPR 2004, Workshop on Generative-Model Based Vision (2004)
14. Zehnder, P., Meier, E.K., Gool, L.V.: An efficient shared multi-class detection cascade. In: BMVC (2008)
15. Nistér, D., Stewénus, H.: Scalable recognition with a vocabulary tree. In: CVPR, vol. 2, pp. 2161–2168 (2006)
16. Weber, M., Welling, M., Perona, P.: Towards automatic discovery of object categories. In: CVPR, pp. 2101–2108 (2000)
17. Gavrila, D.M.: A bayesian, exemplar-based approach to hierarchical shape matching. *IEEE PAMI* 29, 1408–1421 (2007)
18. Moreels, P., Perona, P.: A probabilistic cascade of detectors for individual object recognition. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 426–439. Springer, Heidelberg (2008)
19. Amit, Y., Geman, D., Fan, X.: A coarse-to-fine strategy for multiclass shape detection. *PAMI* 26, 1606–1621 (2004)
20. Gangaputra, S., Geman, D.: A design principle for coarse-to-fine classification. In: CVPR, pp. 1877–1884 (2006)
21. Zhu, L., Lin, C., Huang, H., Chen, Y., Yuille, A.: Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 759–773. Springer, Heidelberg (2008)
22. Fleuret, F., Geman, D.: Coarse-to-fine face detection. *IJCV* 41, 85–107 (2001)
23. Ommer, B., Malik, J.: Multi-scale object detection by clustering lines. In: ICCV (2009)
24. Russell, B., Torralba, A., Murphy, K., Freeman, W.T.: Labelme: a database and web-based tool for image annotation. *IJCV* 77, 157–173 (2008)

# Object Classification Using Heterogeneous Co-occurrence Features

Satoshi Ito and Susumu Kubota

Corporate Research and Development Center, Toshiba Corporation, Japan  
satoshi13.ito@toshiba.co.jp

**Abstract.** Co-occurrence features are effective for object classification because observing co-occurrence of two events is far more informative than observing occurrence of each event separately. For example, a color co-occurrence histogram captures co-occurrence of pairs of colors at a given distance while a color histogram just expresses frequency of each color. As one of such co-occurrence features, CoHOG (co-occurrence histograms of oriented gradients) has been proposed and a method using CoHOG with a linear classifier has shown a comparable performance with state-of-the-art pedestrian detection methods. According to recent studies, it has been suggested that combining heterogeneous features such as texture, shape, and color is useful for object classification. Therefore, we introduce three heterogeneous features based on co-occurrence called color-CoHOG, CoHED, and CoHD, respectively. Each heterogeneous features are evaluated on the INRIA person dataset and the Oxford 17/102 category flower datasets. The experimental results show that color-CoHOG is effective for the INRIA person dataset and CoHED is effective for the Oxford flower datasets. By combining above heterogeneous features, the proposed method achieves comparable classification performance to state-of-the-art methods on the above datasets. The results suggest that the proposed method using heterogeneous features can be used as an off-the-shelf method for various object classification tasks.

## 1 Introduction

Object classification is one of the essential tasks in computer vision and histogram based features such as SIFT (scale invariant feature transform) [9], HOG (histograms of oriented gradients) [11], and a color histogram [17] are widely used features for object classification. A merit of histogram based features is robustness to the slight shift of an object position. However, these histogram based features have the limited discriminative power because they don't take any spatial information into account. One of the solutions to this problem is to extract features from multiple small regions in an image. However, if the regions are too small, features extracted from them become sensitive to the slight object translation. Another solution is to use co-occurrences of pairs of features extracted from different positions in an input image. For example, a color co-occurrence histogram (CCH) [6], also called color correlogram, captures co-occurrence of

pairs of colors while a color histogram just expresses frequency of each color. In a similar way, edge co-occurrence matrices (ECMs) [13], originally applied to texture classification problem, express the spatial relationship of pairs of edge orientations. Recently, CoHOG (co-occurrence histograms of oriented gradients) [19], an extension of HOG to represent the spatial relationship between gradient orientations, has been proposed and its effectiveness for pedestrian detection and cat face detection has been shown in [19,8]. Methods using co-occurrences of more than two features have also been proposed in [20,15].

According to recent studies [4,16,10,11], it has been suggested that combining heterogeneous features such as texture, shape, and color is useful for object classification. Since heterogeneous features represent various aspects of objects and work complementarily, they achieve higher classification performance than homogeneous features and are applicable to a variety of object classification tasks. Therefore, we introduce three heterogeneous features based on co-occurrence called color-CoHOG, CoHED, and CoHD, respectively.

The remainder of the paper is organized as follows. CoHOG is briefly explained in Sect. 2. Then three heterogeneous features, color-CoHOG, CoHED, and CoHD are proposed in Sect. 3. Experiments are presented in Sects. 4 and 5. Finally, conclusions are given in Sect. 6.

## 2 Co-occurrence Histograms of Oriented Gradients

CoHOG (co-occurrence histograms of oriented gradients) [19], an extension of HOG [1], consists of multiple co-occurrence histograms of gradient orientations. Though the dimensionality of CoHOG is high, a linear classifier gives high classification performance. Therefore, computational cost of classification is lower than other complex classification methods such as kernel SVM. An algorithm of CoHOG calculation is shown in Algorithm 1. The number of elements of the co-occurrence histograms  $H$  is  $m \times n \times d^2$  where  $d$  is the number of gradient orientation bins. For example, given 10 offsets, 10 small regions, and 10 bins for gradient orientation, the number of elements of  $H$  is 10,000. In detail, please refer to [19].

## 3 Proposed Features

In this section, we propose three heterogeneous features based on co-occurrence called color-CoHOG, CoHED, and CoHD, respectively. Color-CoHOG, which is an extension of CoHOG to make use of color information, is co-occurrence of a color matching result and a pair of edge directions. CoHED is co-occurrence between edge orientation and color difference. CoHD is co-occurrence of a pair of color differences. Hence color-CoHOG and CoHED are co-occurrences of heterogeneous features and CoHD is co-occurrence of homogeneous features. Details are described in the following sections. We also explain a color histogram as a complementary feature of the above three features.

**Algorithm 1.** CoHOG calculation

---

**Input:**  $I$ : a grayscale image,  $\{(p_i, q_i)\}_{i=1}^m$ :  $m$  offsets,  $\{D_i\}_{i=1}^n$ :  $n$  small regions in the image

- 1: compute a gradient orientation image  $G$  from  $I$
- 2: initialize co-occurrence histograms  $H$  with zeros
- 3: **for**  $i = 1$  to  $m$  **do**
- 4: **for**  $j = 1$  to  $n$  **do**
- 5: **for all**  $(x, y) \in D_j$  **do**
- 6: **if**  $(x + p_i, y + q_i)$  is inside of the image **then**
- 7:  $g_1 \leftarrow G(x, y)$
- 8:  $g_2 \leftarrow G(x + p_i, y + q_i)$
- 9:  $H(g_1, g_2, j, i) \leftarrow H(g_1, g_2, j, i) + 1$
- 10: **end if**
- 11: **end for**
- 12: **end for**
- 13: **end for**
- 14: **return**  $H$

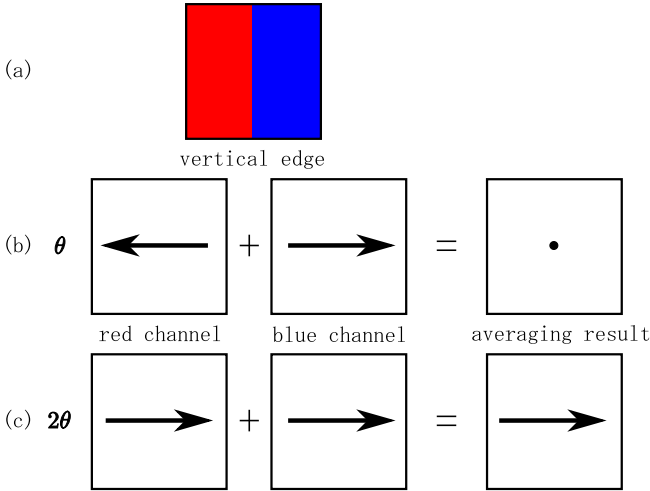
---

### 3.1 Color-CoHOG

CoHOG calculation described in Algorithm 1 assumes that an input image is grayscale. Derivative masks such as Sobel filter are used to compute gradients. If a color image is given, the conversion from color to grayscale is necessary before CoHOG extraction. Therefore, we extend CoHOG to make use of color information and we apply two ideas. First, we calculate edge orientation in a color image instead of a grayscale one. Second, we use a result of color matching in order to take distinction of foreground and background into account. The details of the ideas are described below.

Deciding edge orientation in a color image is not a trivial problem and a lot of researches have been done [7, 14, 12]. We found that a method based on the double angle representation [5] gives the consistent results with reasonable computational cost. In the double angle representation, the directions  $\theta$  and  $\theta + 180$  degrees are equivalent and the orthogonal directions  $\theta$  and  $\theta + 90$  degrees are the vectors that point in opposite directions so that averaging gradients in different color channels makes sense (shown in Fig. 1). As a result, we obtain gradient orientations between 0 and 180 degrees since we make no distinction between  $\theta$  and  $\theta + 180$  degrees. In the experiments described in Sects. 4 and 5, Roberts filter is used to calculate initial gradients and then they are averaged in the double angle representation over the RGB channels and the spatial regions of  $2 \times 2$  pixel size. Averaged gradient orientation is evenly divided into 4 bins.

Foreground-background discrimination is helpful to describe a shape (e.g., [12]). Taking this into account, we use a result of color matching between a pair of pixels at a given offset. This is based on the assumption that colors of a pair of pixels belonging to the same object are likely to be similar while colors of a pair of pixels located at different objects are likely to be dissimilar. In particular, we calculate two co-occurrence histograms per offset and small region, one is the



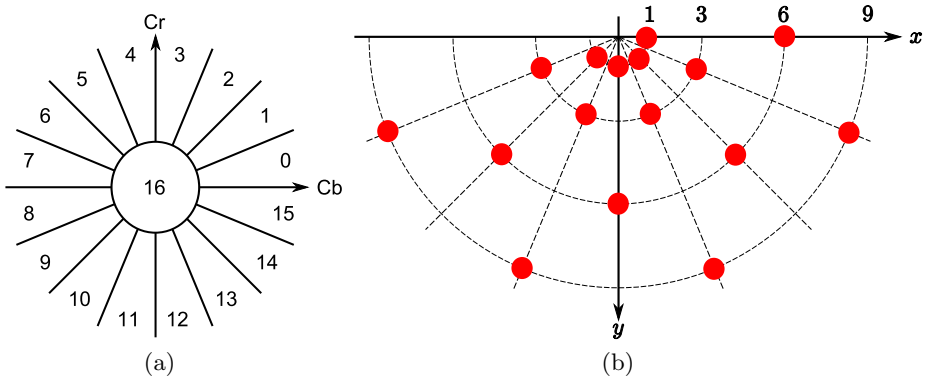
**Fig. 1.** (a) A vertical edge. (b) Averaging gradients, denoted by arrows, over red and blue channels in the single angle representation gives undesirable result. (c) Averaging gradients over red and blue channels in the double angle representation gives desirable result.

co-occurrence histogram of a pair of pixels at a given offset that have the *same* color and the other is the one of a pair of pixels that have *different* colors. For the computational efficiency, we quantize colors in Cb-Cr space into 17 clusters shown in Fig. 2a and compare the cluster labels to decide if a pair of pixels has the same color.

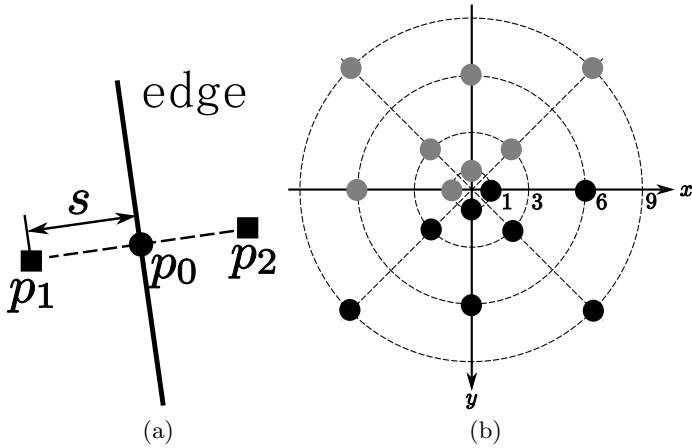
Our proposed feature named color-CoHOG is summarized in Algorithm 2. Whereas the original CoHOG captures texture information only, color-CoHOG can capture both texture and shape information since foreground-background discrimination is taken into account. The dimension of color-CoHOG is  $m \times n \times 2 \times d^2$  where  $d$  is the number of quantized edge directions. In the experiments, since we use 16 offsets shown in Fig. 2b, color-CoHOG has  $16 \times 1 \times 2 \times 4^2 = 512$  elements per small region.

### 3.2 CoHED

We propose a feature CoHED (Co-occurrence Histograms of pairs of Edge orientations and color Differences) that expresses the relationships between an edge orientation and the change of colors across the edge. Once an edge orientation at the point  $p_0$  is determined, two points  $p_1$  and  $p_2$  are located at the two opposite sides of the edge point  $p_0$  (shown in Fig. 3a). Edge orientations are calculated in the same manner as described in Sect. 3.1 and color differences between  $p_1$  and  $p_2$  are calculated in YCbCr color space. Then color differences are quantized to 8 directions in each color plane, that is, Y-Cb plane, Y-Cr plane, and Cb-Cr plane. Calculation of a co-occurrence histogram with color difference in Y-Cb



**Fig. 2.** (a) The figure shows color labels in Cb-Cr space. The label 16 corresponds to neutral gray. (b) The figure shows 16 offsets (drawn as *filled circles*) used for color-CoHOG calculation.



**Fig. 3.** (a) Positions of three points  $p_0, p_1$  and  $p_2$  used for CoHED. Once edge orientation at  $p_0$  is decided,  $p_1$  and  $p_2$  are located at the two opposite sides of the edge. (b) Eight offsets used for CoHD. A set of three pixels that consist of an origin and a pair of points located at two opposite positions with respect to the origin is used to calculate color difference.

plane is as follows;

$$H_{Y-Cb}(g, c) \leftarrow H_{Y-Cb}(g, c) + |dy| + |du| \tag{1}$$

where  $g$  is the edge orientation at  $p_0$ ,  $c$  is the quantized color difference between  $p_1$  and  $p_2$  in Y-Cb plane, and  $dy$  and  $du$  are differences between  $p_1$  and  $p_2$  in Y channel and Cb channel, respectively. CoHED is computed by weighted voting ( $|dy| + |du|$  in (1) corresponds to a voting weight) while other co-occurrence

---

**Algorithm 2.** color-CoHOG calculation

---

**Input:**  $I$ : a color image,  $\{(p_i, q_i)\}_{i=1}^m$ :  $m$  offsets,  $\{D_i\}_{i=1}^n$ :  $n$  small regions in the image

- 1: compute an edge direction image  $G$  from  $I$  using the double angle representation
- 2: compute color labels  $C$  of pixels in the image
- 3: initialize co-occurrence histograms  $H$  with zeros
- 4: **for**  $i = 1$  to  $m$  **do**
- 5:     **for**  $j = 1$  to  $n$  **do**
- 6:         **for all**  $(x, y) \in D_j$  **do**
- 7:             **if**  $(x + p_i, y + q_i)$  is inside of the image **then**
- 8:                  $g_1 \leftarrow G(x, y)$
- 9:                  $g_2 \leftarrow G(x + p_i, y + q_i)$
- 10:                 **if**  $C(x, y)$  is equal to  $C(x + p_i, y + q_i)$  **then**
- 11:                      $c \leftarrow 1$
- 12:                 **else**
- 13:                      $c \leftarrow 0$
- 14:                 **end if**
- 15:                  $H(g_1, g_2, c, j, i) \leftarrow H(g_1, g_2, c, j, i) + 1$
- 16:             **end if**
- 17:         **end for**
- 18:     **end for**
- 19: **end for**
- 20: **return**  $H$

---

features described in this paper are computed by unweighted voting. Since voting weights for strong step-edges are larger than those for weak ones, CoHED mainly captures shape information rather than texture information. We use 1, 3, 6, and 9 as the distance  $s$  from  $p_0$  to  $p_1(p_2)$  in the experiments. Thus, the dimension of CoHED is 4 (edge directions)  $\times$  8 (directions of color differences)  $\times$  3 (color planes)  $\times$  4 (scales) = 384.

### 3.3 CoHD

Since color-CoHOG captures shape and texture information and CoHED captures shape information, it's expected that features mainly capturing texture information work complementarily to color-CoHOG and CoHED. Therefore, based on the similar idea as CoHOG, we propose a feature CoHD (Co-occurrence Histograms of color Differences) that simply captures texture information. CoHD represents changes of color values of three pixels located on a given line in an image (shown in Fig. 3b). Color differences are calculated between the centered pixel and the one of the other two pixels, respectively. Calculation of CoHD is described in Algorithm 3. Color differences in Cb-Cr plane are quantized to 4 directions. Eight offsets (shown in Fig. 3b) are used to calculate color differences of pairs of pixels. Thus, the dimension of CoHD is 4 (directions of color differences)  $\times$  4 (directions of color differences)  $\times$  8 (offsets) = 128.

**Algorithm 3.** CoHD calculation

---

**Input:**  $U$ : Cb-plane image,  $V$ : Cr-plane image,  $\{(p_i, q_i)\}_{i=1}^m$ :  $m$  offsets,  $\{D_i\}_{i=1}^n$ :  $n$  small regions in the image

```

1: initialize co-occurrence histograms  $H$  with zeros
2: for  $i = 1$  to  $m$  do
3:   for  $j = 1$  to  $n$  do
4:     for all  $(x, y) \in D_j$  do
5:       if  $(x + p_i, y + q_i)$  and  $(x - p_i, y - q_i)$  are inside of the image then
6:          $u_1 \leftarrow U(x + p_i, y + q_i) - U(x, y)$ 
7:          $v_1 \leftarrow V(x + p_i, y + q_i) - V(x, y)$ 
8:          $u_2 \leftarrow U(x - p_i, y - q_i) - U(x, y)$ 
9:          $v_2 \leftarrow V(x - p_i, y - q_i) - V(x, y)$ 
10:         $c_1 \leftarrow (u_1 > 0) + 2 \times (v_1 > 0)$  // quantization into 4 directions
11:         $c_2 \leftarrow (u_2 > 0) + 2 \times (v_2 > 0)$  // quantization into 4 directions
12:         $H(c_1, c_2, j, i) \leftarrow H(c_1, c_2, j, i) + 1$ 
13:      end if
14:    end for
15:  end for
16: end for
17: return  $H$ 

```

---

### 3.4 Color Histogram

The above three features use relative color information. However, absolute color information is also useful for object classification [10, 16]. In this paper, we use a simple color histogram that consists of 17 bins shown in Fig. 2a. Since we use a linear classifier in the experiments, 2nd order polynomial terms of elements of a color histogram are explicitly generated in order to increase linear separability. Thus the number of elements including the 2nd order terms is 170.

## 4 Experiment 1. INRIA Person Dataset

In this section, we evaluate the proposed method on the INRIA person dataset [1]. The INRIA person dataset provides positive images cropped  $64 \times 128$  pixels and negative images of various sizes. Some examples are shown in Fig. 4. The number of positive/negative images are 2, 416/1, 218 for training and 1, 132/453 for testing, respectively. Detection performance is evaluated by the same way as described in [1]. We extract features separately from  $4 \times 8$  non-overlapped small regions that are  $16 \times 16$  pixel sizes and concatenate them into a single feature vector. Since the dimensionality of the feature vectors is high, we use a linear classifier trained by LIBLINEAR [3] that is applicable to a large scale problem. Each component of features is normalized by its maximum value in the training samples, respectively.





Fig. 4. Examples in the INRIA person dataset

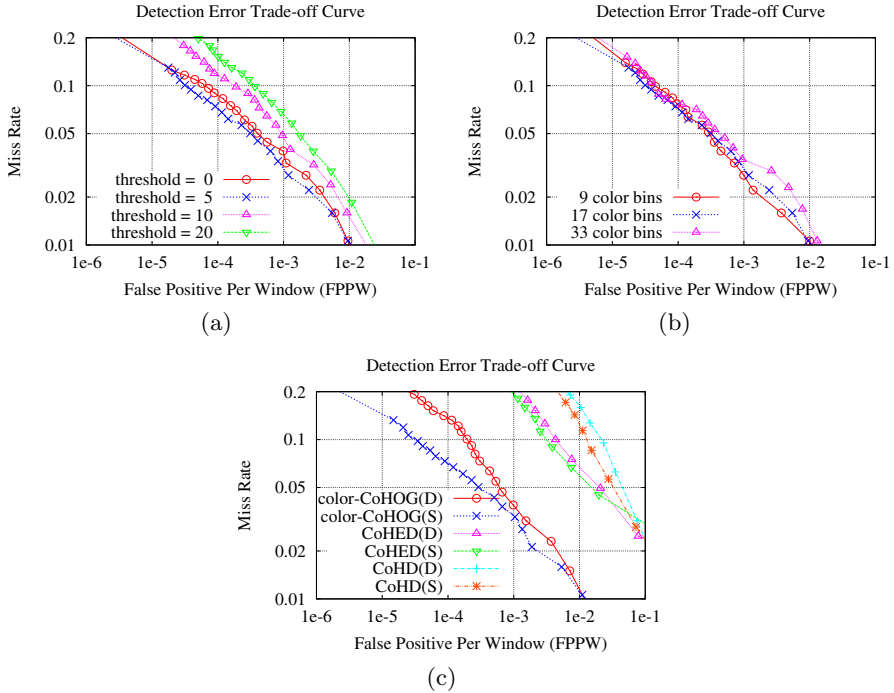


Fig. 5. Feature evaluation on the INRIA person dataset. (a) DET curves of various thresholds for neutral gray. (b) DET curves obtained by changing the number of color bins. (c) DET curves of the sparse setting (denoted by 'S') and the dense setting (denoted by 'D'), respectively.

### 4.1 Feature Evaluation

In this section, we study the effect of the following three parameters; the threshold for neutral gray, the number of color bins, and the scale of the offsets. The former two parameters are related to color-CoHOG and the last parameter is related to color-CoHOG, CoHED and CoHD, respectively. Since the dimensionality of features isn't affected by changing the above three parameters, detection performances obtained by changing those parameters can be easily compared with

each other. On the other hand, the dimensionality of features is proportional to the square of the number of quantized directions, which is another parameter of the proposed features. In this case, it's difficult to compare the detection performances. Thus we select a practical value for the number of quantized directions and it's used in the experiments in this paper.

The threshold for neutral gray is the parameter that decides whether each pixel is chromatic (labels 0-15 in Fig. 2a) or achromatic (label 16 in Fig. 2a) based on the distance from the origin in Cb-Cr space. Figure 5a shows the DET (detection error trade-off) curves obtained by changing the threshold. The experimental result suggests that a small threshold that classifies most of the pixels as chromatic works well. The setting that classifies all the pixels as chromatic also works as well (threshold = 0 in Fig. 5a). We set the threshold to 5 in other experiments described in this paper.

We also studied the effect of the number of color bins. The experimental result shows that the result of 33 color bins is slightly worse than the other two results but the number of color bins is insensitive to the detection performance (shown in Fig. 5b). We use 17 color bins in other experiments described in the paper.

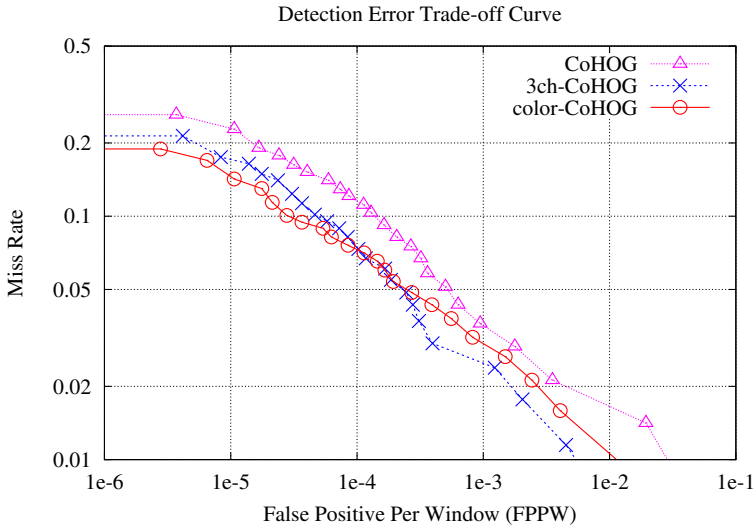
The scale of the offsets is the parameter that decides the distance between the center pixel and the pixel with an offset. We tested two cases; one is a sparse setting and the other is a dense setting. The sparse setting uses four scales 1, 3, 6 and 9 as the distances between pixels in Figs. 2b and 3 while the dense setting uses 1, 2, 3 and 4. The results of color-CoHOG and CoHD show that the sparse setting is better than the dense one and the result of CoHED shows that the sparse setting is a little bit better than the dense one (shown in Fig. 5c). This suggests that capturing less redundant information is more important to improve classification performance. Therefore, the sparse setting is used in other experiments in the paper.

## 4.2 Comparison with CoHOG

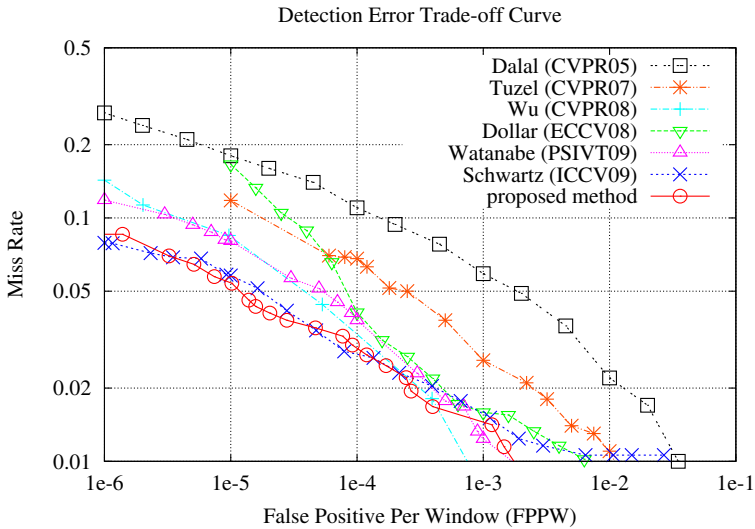
Figure 6 shows the DET curves of CoHOG and color-CoHOG, respectively. We also plotted the result of 3ch-CoHOG as another extension of CoHOG to make use of color information. 3ch-CoHOG is a feature obtained by concatenating CoHOGs extracted separately from each color channel. The offsets that are used for color-CoHOG (shown in Fig. 2b) are used to calculate CoHOG and 3ch-CoHOG for comparison under the same condition. The detection performance of color-CoHOG is superior to that of CoHOG and comparable to that of 3ch-CoHOG while the dimensionality of color-CoHOG (16, 384) is half as that of CoHOG (32, 768) and only one-sixth of that of 3ch-CoHOG (98, 304), respectively. This result means that color-CoHOG makes use of color information efficiently.

## 4.3 Comparison with Previous Methods

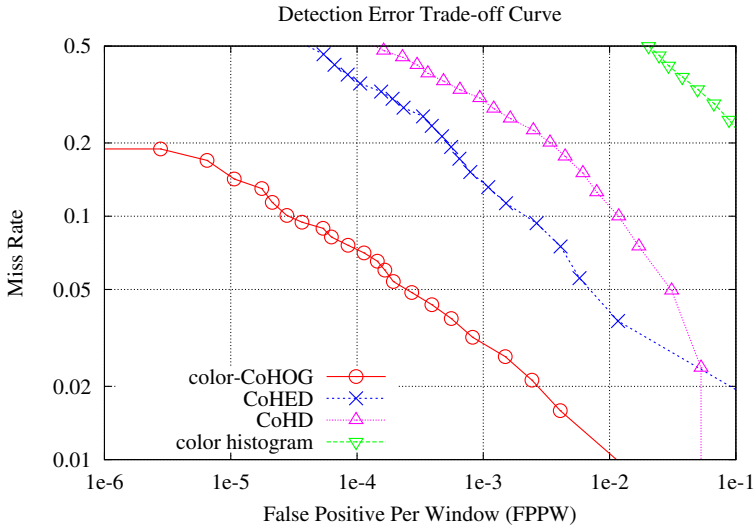
Figure 7 compares the DET curves of the proposed method with those of the previous methods [1, 18, 21, 2, 19, 16]. Four heterogeneous features, color-CoHOG, CoHED, CoHD, and color histograms, were used for the proposed method. The



**Fig. 6.** DET curves of several CoHOGs on the INRIA person dataset. Color-CoHOG (*circle*) is superior to CoHOG (*triangle*) and comparable to 3ch-CoHOG (*cross*) while the dimensionality of color-CoHOG is half as that of CoHOG and only one-sixth of that of 3ch-CoHOG.



**Fig. 7.** DET curves of the proposed method and several previous methods on the INRIA person dataset. This figure shows that the proposed method (*circle*) is comparable to the state-of-the-art method (*cross*).



**Fig. 8.** DET curves of single features on the INRIA person dataset

curves of the previous methods were obtained by tracing the results in the references. The proposed method achieves 8.6%, 5.5% and 2.9% miss rates at  $10^{-6}$ ,  $10^{-5}$  and  $10^{-4}$  FPPWs (false positive per window), respectively. This result is comparable to the state-of-the-art method [16] that has achieved 7.9% miss rate at  $10^{-6}$  FPPW and 5.8% miss rate at  $10^{-5}$  FPPW.

We also show the DET curves of single features in Fig. 8. The result of each single feature except color-CoHOG is far inferior to the method of Dalal et al. [1] (shown in Fig. 7) while the method using the concatenated features achieves comparable performance to the state-of-the-art method as described above. This means that our proposed features provide complementary information to each other.

## 5 Experiment 2. Oxford 17/102 Category Flower Datasets

In this section, we evaluate the proposed method on the Oxford 17/102 category flower datasets [10,11]. The 17 category dataset consists of 80 images per category and the 102 category dataset consists of between 40 and 258 images per category. Some examples are shown in Fig. 9. Classification performance is evaluated by the same manner as described in [10].

In [10], they provide training images, validation images and test images though we don't use validation images since they are not necessary for the proposed method. There are various sizes of images in the datasets, so we crop and resize them into  $64 \times 64$  pixel size. We extract color-CoHOG, CoHED, CoHD, and a color histogram from the whole region of the resized image and concatenate



**Fig. 9.** Examples of the Oxford 17 category flower dataset

**Table 1.** Classification performance on the Oxford flower datasets

Method	Performance score [10]	
	17 categories	102 categories
Nilsback [11]	88.33±0.30	72.8
color-CoHOG+CoHED+CoHD+color histogram	94.19±1.22	74.8
color-CoHOG	78.89±1.19	43.4
CoHED	91.54±0.99	64.2
CoHD	84.24±1.07	57.0
color histogram	69.88±2.68	35.6

them into a single feature vector. The dimension of the resulting feature vector is 1,194. In the same manner as described in Sect. 4, linear classifiers trained by LIBLINEAR are used and each component of features is normalized by its maximum value.

Experimental results are shown in Table 1. The proposed method using all features described in Sect. 3 achieves higher classification performance than the state-of-the-art method [11] on both datasets in spite of the simplicity of the proposed method. CoHED achieves the best classification performance among the four single features on both flower datasets while color-CoHOG achieves the best performance on the INRIA person dataset. This means that effective features are different with respect to object classification tasks. Therefore, a method using homogeneous features, which is effective for a specific object classification task, may fail to achieve high classification performance for another object classification task. In contrast, the proposed method using heterogeneous features can be used as an off-the-shelf method for various object classification tasks.

## 6 Conclusion

In this paper, we proposed three heterogeneous features based on co-occurrence called color-CoHOG, CoHED, and CoHD, respectively and introduced a color

histogram as a complementary feature of those three features. Co-occurrence features are very high dimensional features and highly discriminative, so that a linear classifier is sufficient to achieve high classification performance. Classification performance of each feature was evaluated on the INRIA person dataset and the Oxford 17/102 category flower datasets, respectively. The experimental results show that effective features for the INRIA person dataset are different from those for the Oxford flower datasets. By combining the above four heterogeneous features, the proposed method achieved comparable performance to state-of-the-art methods on the above datasets. The results suggest that the proposed method using heterogeneous features can be used as an off-the-shelf method for various object classification tasks.

## References

1. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 886–893 (2005)
2. Dollár, P., Babenko, B., Belongie, S., Perona, P., Tu, Z.: Multiple component learning for object detection. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 211–224. Springer, Heidelberg (2008)
3. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. *Machine Learning Research* 9, 1871–1874 (2008)
4. Gehler, P.V., Nowozin, S.: On feature combination for multiclass object classification. In: Proceedings of the Twelfth IEEE International Conference on Computer Vision (2009)
5. Granlund, G.H.: In search of a general picture processing operator. In: *Computer Graphics and Image Processing*, pp. 155–173 (1978)
6. Huang, J., Kumar, S.R., Mitra, M., Zhu, W.J., Zabih, R.: Image indexing using color correlograms. In: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition, p. 762. IEEE Computer Society, Washington (1997)
7. Koschan, A.: A comparative study on color edge detection. In: Proceedings of the 2nd Asian Conference on Computer Vision, pp. 574–578 (1995)
8. Kozakaya, T., Ito, S., Kubota, S., Yamaguchi, O.: Cat face detection with two heterogeneous features. In: Proceedings of the 2009 IEEE International Conference on Image Processing (2009)
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision* 60(2), 91–110 (2004)
10. Nilsback, M.E., Zisserman, A.: A visual vocabulary for flower classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 1447–1454 (2006)
11. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing (December 2008)
12. Ott, P., Everingham, M.: Implicit color segmentation features for pedestrian and object detection. In: Proceedings of the Twelfth IEEE International Conference on Computer Vision (2009)
13. Rautkorpi, R., Iivarinen, J.: A novel shape feature for image classification and retrieval. In: Campilho, A.C., Kamel, M.S. (eds.) ICIAR 2004, Part I. LNCS, vol. 3211, pp. 753–760. Springer, Heidelberg (2004)

14. Ruzon, M.A., Tomasi, C.: Color edge detection with the compass operator. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 160–166 (1999)
15. Sabzmeydani, P., Mori, G.: Detecting pedestrians by learning shapelet features. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
16. Schwartz, W.R., Kembhavi, A., Harwood, D., Davis, L.S.: Human detection using partial least squares analysis. In: Proceedings of the Twelfth IEEE International Conference on Computer Vision (2009)
17. Swain, M.J., Ballard, D.H.: Color indexing. *Int. Journal of Computer Vision* 7(1), 11–32 (1991)
18. Tuzel, O., Porikli, F., Meer, P.: Human detection via classification on riemannian manifolds. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2007)
19. Watanabe, T., Ito, S., Yokoi, K.: Co-occurrence histograms of oriented gradients for pedestrian detection. In: The 3rd Pacific Rim Symposium on Advances in Image and Video Technology, pp. 37–47 (2009)
20. Wu, B., Nevatia, R.: Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In: The Tenth IEEE International Conference on Computer Vision, vol. 1, pp. 90–97. IEEE Computer Society, Washington (2005)
21. Wu, B., Nevatia, R.: Optimizing discrimination-efficiency tradeoff in integrating heterogeneous local features for object detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2008)

# Converting Level Set Gradients to Shape Gradients

Siqi Chen<sup>1,\*</sup>, Guillaume Charpiat<sup>2</sup>, and Richard J. Radke<sup>1</sup>

<sup>1</sup> Department of ECSE, Rensselaer Polytechnic Institute, Troy, NY, USA  
chens@rpi.edu, rjradke@ecse.rpi.edu

<sup>2</sup> Pulsar Project, INRIA Sophia-Antipolis, France  
Guillaume.Charpiat@sophia.inria.fr

**Abstract.** The level set representation of shapes is useful for shape evolution and is widely used for the minimization of energies with respect to shapes. Many algorithms consider energies depending explicitly on the signed distance function (SDF) associated with a shape, and differentiate these energies with respect to the SDF directly in order to make the level set representation evolve. This framework is known as the “variational level set method”. We show that this gradient computation is actually mathematically incorrect, and can lead to undesirable performance in practice. Instead, we derive the expression of the gradient with respect to the shape, and show that it can be easily computed from the gradient of the energy with respect to the SDF. We discuss some problematic gradients from the literature, show how they can easily be fixed, and provide experimental comparisons illustrating the improvement.

## 1 Introduction

In recent years, much work on geometric active contour models, i.e. active contour models [1] implemented with the level set method [2], has been proposed to solve many computer vision problems [3]. Any planar closed curve  $\Gamma$ , i.e. any function  $\Gamma : \mathbb{S}^1 \rightarrow \Omega$  from the circle  $\mathbb{S}^1$  to the image domain  $\Omega \subset \mathbb{R}^2$ , can be represented by the zero level set of a higher-dimensional embedding function  $\phi : \Omega \rightarrow \mathbb{R}$ . During curve evolution, instead of directly updating the contour  $\Gamma$ , one can then update its associated embedding function  $\phi$ , which is more practical for handling topological changes, like merging or splitting. The embedding function  $\phi$  is usually the signed distance function (SDF) of  $\Gamma$ , i.e. the function that associates any point  $x \in \Omega$  with the signed distance  $\phi(x) = \pm d(x, \Gamma)$  from  $x$  to  $\Gamma$ , with a minus sign if  $x$  belongs to the interior of the region delimited by  $\Gamma$ .

Many computer vision problems involving shapes can be formulated as the minimization of a certain energy functional  $E(\Gamma)$ . Depending on the properties of the energy  $E$ , one is often reduced to performing gradient descents with respect

---

\* This work is supported in part by CenSSIS, the NSF Center for Subsurface Sensing and Imaging Systems, under the award EEC-9986821.



to the shape  $\Gamma$ , starting from an initialization  $\Gamma_0$  and making  $\Gamma$  evolve step by step in the opposite direction of the gradient of the energy:

$$\begin{cases} \Gamma(0) &= \Gamma_0 \\ \frac{\partial \Gamma(t)}{\partial t} &= -\nabla_{\Gamma} E(\Gamma(t)) \end{cases} \tag{1}$$

where the gradient  $\nabla_{\Gamma} E(\Gamma)$  is defined from the derivative of  $E$  with respect to  $\Gamma$  and depends on the choice of an inner product (see Section 3.1 for a proper definition, or 4). If one chooses to represent the contour  $\Gamma(t)$  by an embedding function  $\phi(t)$ , then one is interested in the equation that governs the evolution of  $\phi(t)$ . Since by definition  $\forall t, \forall x \in \Gamma(t), \phi(t)(x) = 0$ , one obtains, by differentiation,  $\frac{\partial \phi}{\partial t} + \frac{\partial \phi}{\partial x} \frac{\partial \Gamma}{\partial t} = 0$ , and thus:

$$\begin{cases} \phi(0) &= \phi_0 \text{ (e.g. := SDF}(\Gamma_0) \text{)} \\ \frac{\partial \phi}{\partial t}(x) &= |\nabla_x \phi(t)|(x) V(t)(x) \quad \forall x \in \Gamma(t) = \phi(t)^{-1}(0) \end{cases} \tag{2}$$

where  $\forall x \in \Gamma, V(t)(x) = \nabla_{\Gamma} E(\Gamma(t))(x) \cdot \mathbf{n}_{\Gamma(t)}(x)$  is the normal velocity field, i.e. the part of the shape gradient that is normal to the contour  $\Gamma$ . This so-called ‘‘level set equation’’ has to be properly defined for points which are not on  $\Gamma$ , e.g. by extending the velocity field  $V$  to at least a narrow band around  $\Gamma$ .

However, for many applications, calculating the gradient using Eq. (1) directly is difficult and therefore an alternative derivation to obtain the level-set equation was proposed in 5, named the ‘‘variational level set method’’. The idea is that since there is a bijection between  $\Gamma$  and its signed distance function  $\phi$ , an energy defined on shapes  $E(\Gamma)$  can be rewritten as an energy  $F(\phi)$  defined on their level set representations and vice versa. Subsequently, one might be interested in deriving the Euler-Lagrange equation that minimizes  $F(\phi)$  directly:

$$\frac{\partial \phi}{\partial t} = -\nabla_{\phi} F(\phi) =: G(\phi). \tag{3}$$

Note that this equation is *a priori* not related to Eq. (2). Since the introduction of the variational level set approach, much work has been carried out under this framework, e.g. 6,7,8, to name a few. However only very few functions in  $L^2(\Omega \rightarrow \mathbb{R})$  are SDFs of a shape; for instance, they must satisfy the Eikonal Equation  $|\nabla_x \phi| = 1$  almost everywhere. Therefore, a new  $\phi$  obtained from a discrete step of Eq. (3) will generally not itself be a valid SDF. Various authors 5,9,10 showed that if the velocity field  $G$  in Eq. (3) satisfies

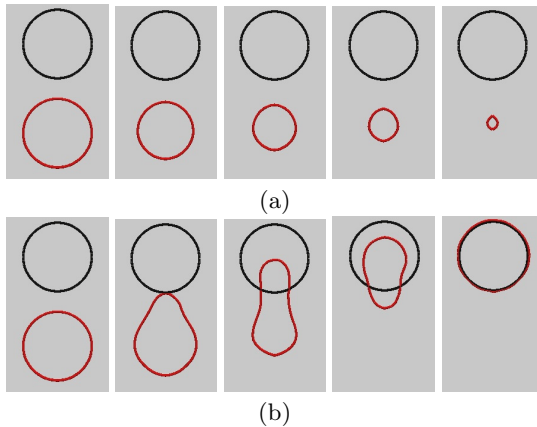
$$\nabla_x G \cdot \nabla_x \phi = 0 \tag{4}$$

then the evolving level set  $\phi$  will always remain a valid SDF for all time. However, a general level set gradient  $G = -\nabla_{\phi} F(\phi)$  is unlikely to satisfy Eq. (4), and several approaches have been proposed to maintain the SDF property of  $\phi$  11.

The first approach is known as “velocity extension” [9,10]. A narrow band around  $\Gamma$  is first defined, and Eq. (4) is solved to extend  $G$  with the “Fast Marching” method [10]. A second approach is known as “reinitialization” [12]. Here, Eq. (3) is used to update  $\phi$ , but since the newly obtained  $\phi$  will drift away from a SDF, the evolution is occasionally stopped, and the SDF of the zero level set of  $\phi$  is recomputed. A third approach was proposed by Li et al. [13], in which the deviation of  $|\nabla_x \phi|$  from 1 is incorporated into the energy function  $F(\phi)$ .

However, previous works neglect one important issue. Although the desired energy  $E(\Gamma)$  can be rewritten as  $F(\phi)$ , the meaning of the gradient  $\nabla_\phi F(\phi)$  is fundamentally different from  $\nabla_\Gamma E(\Gamma)$  because the computation  $\nabla_\phi F(\phi)$  is performed without the constraint that the gradient should belong to the very particular subset of variations of  $\phi$  that maintain its property of being a SDF. Thus the effect of the gradient  $\nabla_\phi F(\phi)$  on the zero level  $\Gamma$  may be completely different from  $\nabla_\Gamma E(\Gamma)$  and entirely incorrect. Updating the level set function to maintain the SDF property (e.g. by recomputing it from its zero level) does not change the fact that the newly obtained  $\Gamma$  associated with  $\phi$  is wrong.

In this work, we show that with a simple “velocity projection” step, the level set gradient can be made to exactly match the true gradient of  $E(\Gamma)$  with respect to  $\Gamma$ , which we call the “shape gradient”. Therefore, with our approach, one can still take the derivative  $\nabla_\phi F(\phi)$  of an energy  $F(\phi)$  with respect to  $\phi$ , and transform it to obtain the correct shape gradient to deform  $\phi$ . To motivate the discussion, Fig 1 compares our result with the standard variational level set method, where  $E(\Gamma)$  is defined as the  $L^2$  distance from  $\phi$  to the SDF  $\phi_T$  corresponding to a target curve  $\Gamma_T$ , i.e.  $F(\phi) = \|\phi - \phi_T\|_{L^2}^2$ . We can see that using the standard (incorrect) level set method, the initial curve would shrink to a point, while with our corrected gradient, the initial curve is correctly drawn to  $\Gamma_T$ . For more detailed discussion, please refer to Section 4.



**Fig. 1.** Curve evolution of the red circle  $\Gamma$ , where the black circle  $\Gamma_T$  is the target. The cost function is the  $L^2$  distance between the two SDFs. (a) standard variational level set method, (b) our result.

This paper is organized as follows. In Section 2 we describe the family of admissible SDF variations  $\delta\phi$  so that  $\phi + \delta\phi$  remains a valid SDF. In Section 3 we use this family to draw the connection between the level set gradient obtained from Eq.(3) and the shape gradient from Eq.(11). We show that by projecting the level set gradient onto the family of admissible SDF variations, we exactly recover the shape gradient. Therefore, with only a simple “velocity projection” step, we can convert any level set gradient to the correct shape gradient and thus deform  $\phi$  by extending the shape gradient with the usual velocity extension approach. We also show how to directly compute the correct deformation of  $\phi$  by integrating the level set gradient over parts of the image  $\Omega$ , without explicitly computing the shape gradient. In Section 4 we review some often-used level set gradients from the literature that need the “velocity projection” step to make them correct deformations of  $\Gamma$ . We also show experimental results that compare the corrected shape gradient with problematic level set gradients. We then conclude and discuss future directions.

## 2 The Family of Admissible SDF Variations

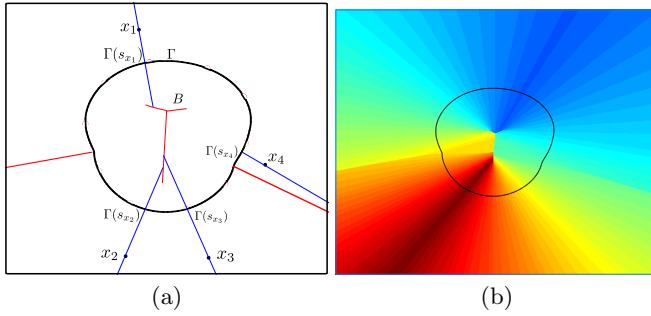
Let us consider a closed planar curve  $\Gamma \in L^2(\mathbb{S}^1 \rightarrow \Omega)$ , an infinitesimal deformation field  $\delta\Gamma \in L^2(\Gamma \rightarrow \mathbb{R}^2)$ , which can be seen as a function in  $L^2(\mathbb{S}^1 \rightarrow \mathbb{R}^2)$  using  $\Gamma$ 's parameterization, and let  $\phi \in L^2(\Omega \rightarrow \mathbb{R})$  be the SDF associated with  $\Gamma$ . If we consider an infinitesimal variation  $\delta\phi$  of  $\phi$ , which is any function in  $L^2(\Omega \rightarrow \mathbb{R})$ , in the general case  $\phi + \delta\phi$  would not be a valid SDF of some corresponding shape. Thus we should only consider variations  $\delta\phi$  so that there exists a shape  $\Gamma'$  so that  $\phi + \delta\phi$  is the SDF of it, i.e.  $\phi + \delta\phi = \Phi(\Gamma')$ .

Let us call  $\mathcal{F}$  the family of all such infinitesimal deformations  $\delta\phi$ . There is a bijection between SDF variations  $\delta\phi$  in  $\mathcal{F}$  and shape deformations  $\delta\Gamma$  of  $\Gamma$ ; that is, for any vector field  $\delta\Gamma$  normal to  $\Gamma$  at each point of  $\Gamma$  (since tangent displacements do not affect the shape), we can associate a corresponding SDF variation  $\delta\phi$  and vice versa. We show in the appendix that to match the shape deformation  $\delta\Gamma$ , one has to update  $\delta\phi$  according to:

$$\forall x \in \Omega, \quad \delta\phi(x) = -\delta\Gamma(s_x) \cdot \mathbf{n}_{\Gamma(s_x)} \quad (5)$$

where  $\Gamma(s_x)$  is the projection of point  $x$  onto  $\Gamma$ , and  $\mathbf{n}_{\Gamma(s_x)}$  is the unit normal at point  $\Gamma(s_x)$  pointing outwards. Here  $\delta\phi$  can be understood as  $\frac{d\phi}{d\Gamma}(\delta\Gamma)$ . Note however that  $\frac{d\phi}{d\Gamma}$  is not defined when a topological change occurs, so that  $\phi$  will have to be recomputed from its 0-level after topological changes.

Fig. 2 illustrates the admissible variations of an SDF. Intuitively, Eq.(5) implies that a valid deformation  $\delta\phi$  at any point  $x$  in  $\Omega$  depends only on its projection onto  $\Gamma$ : if two points share the same projection point  $\Gamma(s_x)$ , then their variation will be the same. This is a known result [5,9,10]. Consequently, all points on a projection line vary the same way, i.e.  $\delta\phi$  is a constant along projection lines to  $\Gamma$ . Conversely, if  $\delta\phi$  is constant along all projection lines to  $\Gamma$ , then there exists a deformation  $\delta\Gamma$  associated with it. Note that the projection



**Fig. 2.** Illustration of the admissible variations of a SDF. (a) The projection line (in blue) is the line going through  $x$  that orthogonally intersects  $\Gamma$  at  $\Gamma(s_x)$  and stops at the skeleton  $B$ . (b) The admissible variation  $\delta\phi$  is a constant along the projection lines.

$\Gamma(s_x)$  is well-defined for all points in  $\Omega$  except for those on the skeleton of  $\Gamma$ . Since we will later integrate bounded variations over regions of  $\Omega$  in which the Lebesgue measure of the skeleton is 0, this will pose no problem in our work.

As a consequence, the family  $\mathcal{F}$  of all admissible variations of a SDF  $\phi$  is the set of all  $L^2(\Omega \rightarrow \mathbb{R})$  functions that are constant along projection lines to  $\Gamma$ . This means that, when performing a shape evolution based on a level set representation, one should ensure that the level set variation belongs to this family  $\mathcal{F}$ . Numerical algorithms such as the Fast Marching method [10] can be used to obtain such a level set variation based on the deformation of  $\Gamma$ .

### 3 Velocity Projection

In the previous section, we defined the family of all admissible variations of an SDF as the set of all functions that are constant along projection lines to  $\Gamma$ . We will now use this result to draw a connection between  $\nabla_\phi F(\phi)$ , which we call the “level set gradient” and  $\nabla_\Gamma E(\Gamma)$ , which we call the “shape gradient”. We will show that projecting  $\nabla_\phi F(\phi)$  onto the family  $\mathcal{F}$  will exactly produce  $\nabla_\Gamma E(\Gamma)$ .

#### 3.1 Gradients and Inner Products

The gradient definition depends on the choice of the inner product in the tangent space of shapes [4,15]. In this work, we use the standard  $L^2$  inner product:

$$\langle \delta\Gamma_1 | \delta\Gamma_2 \rangle_{L^2(\mathbb{S}^1 \rightarrow \mathbb{R})} = \int_\Gamma \delta\Gamma_1(s) \cdot \delta\Gamma_2(s) \, d\Gamma(s) \tag{6}$$

where  $\delta\Gamma_1$  and  $\delta\Gamma_2$  are two deformations of  $\Gamma$ , where  $s$  denotes a parameterization of  $\Gamma$ , and where  $d\Gamma(s) = \left\| \frac{d\Gamma}{ds} \right\|_{\mathbb{R}^2} ds$  is the associated differential element (i.e. its parameterization-independent length). The gradient associated with this inner product is then defined as the unique deformation  $\nabla_\Gamma E(\Gamma)$  that satisfies

$$\forall \delta\Gamma, \quad DE(\Gamma)(\delta\Gamma) = \langle \nabla_\Gamma E(\Gamma) | \delta\Gamma \rangle_{L^2(\mathbb{S}^1 \rightarrow \mathbb{R})} \tag{7}$$

where  $DE(\Gamma)(\delta\Gamma)$  is the usual directional derivative of  $E$  at  $\Gamma$  along the direction  $\delta\Gamma$ . One of our goals for future work (discussed in the last section) is to extend our current framework to other inner products.

### 3.2 Relating the Two Gradients

Since  $\phi$  is a function of  $\Gamma$  (its SDF), and since  $E(\Gamma) = F(\phi(\Gamma)) \forall \Gamma$ , we have  $DE(\Gamma)(\delta\Gamma) = DF(\phi)(\frac{d\phi}{d\Gamma}(\delta\Gamma))$ , i.e.  $DE(\Gamma)(\delta\Gamma) = DF(\phi)(\delta\phi)$ . On the one side:

$$DE(\Gamma)(\delta\Gamma) = \langle \nabla_{\Gamma} E(\Gamma) | \delta\Gamma \rangle_{L^2(\mathbb{S}^1 \rightarrow \mathbb{R})}$$

while on the other side, Eq. (5) and the definition of the gradient in Eq. (7) give:

$$DF(\phi)(\delta\phi) = \langle \nabla_{\phi} F(\phi) | \delta\phi \rangle_{L^2(\Omega \rightarrow \mathbb{R})} = \langle \nabla_{\phi} F(\phi) | -\delta\Gamma(s_x) \cdot \mathbf{n}_{\Gamma(s_x)} \rangle_{L^2(\Omega \rightarrow \mathbb{R})}$$

so that combining both sides gives:

$$- \int_{\Omega} \nabla_{\phi} F(\phi)(x) (\delta\Gamma(s_x) \cdot \mathbf{n}_{\Gamma(s_x)}) dx = \int_{\Gamma} \nabla_{\Gamma} E(\Gamma)(s) \cdot \delta\Gamma(s) d\Gamma(s) \quad (8)$$

We are now ready to derive a more explicit relation between these two gradients.

As we pointed out, the Lebesgue measure of the skeleton is 0, and under smoothness assumptions about  $F$ , the integrand is bounded and consequently the integral over  $\Omega$  is the same as the integral over  $\Omega \setminus \text{Skeleton}(\Gamma)$ . We also note that any point  $x \in \Omega \setminus \text{Skeleton}(\Gamma)$  not on the skeleton can be written as:

$$x = \Gamma(s_x) + \phi(x) \mathbf{n}_{\Gamma(s_x)} = \Gamma(s) + r \mathbf{n}_{\Gamma(s)}$$

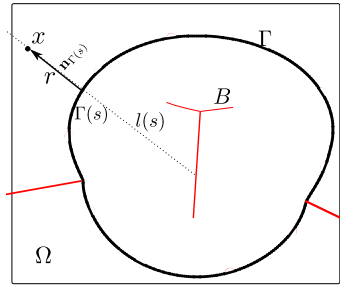
where  $s(x) = s_x$  and  $r(x) = \phi(x)$ , which is illustrated in Fig. 3. Note that  $r$  can be negative. We will define a new coordinate system using  $s$  and  $r$  such that the mapping that associates  $x \in \Omega \setminus \text{Skeleton}(\Gamma)$  with  $(s, r)$  is injective. The infinitesimal (vector) elements of the two coordinate systems are related by:

$$\begin{aligned} d\mathbf{x} &= \left\| \frac{d\Gamma}{ds} \right\|_{\mathbb{R}^2} \mathbf{t}_{\Gamma(s)} ds + \mathbf{n}_{\Gamma(s)} dr - r \kappa_{\Gamma(s)} \left\| \frac{d\Gamma}{ds} \right\|_{\mathbb{R}^2} \mathbf{t}_{\Gamma(s)} ds \\ &= (1 - \kappa_{\Gamma(s)} r) \left\| \frac{d\Gamma}{ds} \right\|_{\mathbb{R}^2} \mathbf{t}_{\Gamma(s)} ds + \mathbf{n}_{\Gamma(s)} dr \end{aligned}$$

where  $\mathbf{t}_{\Gamma(s)}$  is the unit tangent of  $\Gamma(s)$ , and  $\kappa_{\Gamma(s)}$  is the curvature of  $\Gamma(s)$ , which means by definition  $\frac{d}{ds} \Gamma(s) = \left\| \frac{d\Gamma}{ds} \right\|_{\mathbb{R}^2} \mathbf{t}_{\Gamma(s)}$  and  $\frac{d}{ds} \mathbf{n}_{\Gamma(s)} = -\kappa_{\Gamma(s)} \left\| \frac{d\Gamma}{ds} \right\|_{\mathbb{R}^2} \mathbf{t}_{\Gamma(s)}$ .

The determinant of the Jacobian  $\left| \frac{dx}{ds}, \frac{dx}{dr} \right|$ , which is the ratio between the infinitesimal area elements, is then  $|1 - \kappa_{\Gamma(s)} r| \left\| \frac{d\Gamma}{ds} \right\|_{\mathbb{R}^2}$ . Therefore, the right side of Eq. (8) can be rewritten as:

$$\begin{aligned} & - \int_{\Omega} \nabla_{\phi} F(\phi)(x) (\delta\Gamma(s_x) \cdot \mathbf{n}_{\Gamma(s_x)}) dx \\ &= - \int_{\Omega} \nabla_{\phi} F(\phi)(x_{(s,r)}) (\delta\Gamma(s) \cdot \mathbf{n}_{\Gamma(s)}) |1 - \kappa_{\Gamma(s)} r| \left\| \frac{d\Gamma}{ds} \right\|_{\mathbb{R}^2} dr ds \\ &= - \int_{\Gamma} \int_{l(s)} \nabla_{\phi} F(\phi)(x_{(s,r)}) |1 - \kappa_{\Gamma(s)} r| dr \mathbf{n}_{\Gamma(s)} \cdot \delta\Gamma(s) d\Gamma(s) \end{aligned}$$



**Fig. 3.** Illustration of the change of coordinates. See text for explanations.

where  $l(s)$  is the projection line that goes through  $\Gamma(s)$  as illustrated in Fig. 3. It is the set of all points of  $\Omega$  whose projection on  $\Gamma$  is  $\Gamma(s)$ . It is thus a part of a line, which stops at the skeleton of  $\Gamma$  and is also delimited by the boundary of  $\Omega$ . Therefore, a projection line  $l(s)$  is a segment and the integral is well-defined. Since the equality Eq. (8) holds for all possible shape deformations  $\delta\Gamma$ , we obtain:

$$\nabla_{\Gamma} E(\Gamma)(s) = - \int_{l(s)} \nabla_{\phi} F(\phi)(x_{(s,r)}) |1 - \kappa_{\Gamma(s)} \phi(x_{(s,r)})| dr \mathbf{n}_{\Gamma(s)} \quad (9)$$

with  $\phi(x_{(s,r)})$  being just  $r$  by definition. This is the key contribution of our work, since it draws the connection between the shape gradient  $\nabla_{\Gamma} E(\Gamma)$  and the level set gradient  $\nabla_{\phi} F(\phi)$  frequently used in the literature. The intuitive explanation of Eq. (9) is that the shape gradient  $\nabla_{\Gamma} E(\Gamma)$  at  $\Gamma(s)$  is a weighted integral of the level set gradient  $\nabla_{\phi} F(\phi)$  along the projection line going through  $\Gamma(s)$ . We will shortly introduce a natural interpretation of these weights.

### 3.3 The Correct Way to Evolve the Level Sets

Eq. (9) shows how to calculate the shape gradient  $\nabla_{\Gamma} E(\Gamma)$  from the level set gradient  $\nabla_{\phi} F(\phi)$ . However, to actually update the level sets, we need to find the corresponding variation of  $\phi$ , that is  $\delta\phi$ . One possibility for this is to use the classical “velocity extension” approach [9,10] where the velocity defined on  $\Gamma$  is extended to  $\Omega$ . This involves the computation of the zero level set, which is sometimes undesirable. Another way is to directly express  $\delta\phi$  using Eq. (5):

$$\begin{aligned} \delta\phi(x) &= -\delta\Gamma(s_x) \cdot \mathbf{n}_{\Gamma(s_x)} \\ &= - \int_{l(s_x)} |1 - \kappa(s_x) \phi(x'_{(s_x,r)})| \nabla_{\phi} F(\phi)(x'_{(s_x,r)}) dr \end{aligned} \quad (10)$$

To compute the variation  $\delta\phi(x)$  at point  $x$ , it is thus sufficient to integrate the level set gradient, weighted by the area element ratio, along the projection line  $l(s_x)$  that shares the same projection point as  $x$ . We will now give a natural geometrical interpretation of Eq. (9) and Eq. (10) which will lead to our numerical implementation algorithm.

### 3.4 Implementation

Let us first examine the term  $|1 - r \kappa_{\Gamma(s)}|$  in Eq.(9). From basic differential geometry, we have that  $\kappa_{\Gamma(s)} = \frac{d\theta}{ds} \left\| \frac{d\Gamma(s)}{ds} \right\|^{-1} = \frac{1}{R}$ , where  $R$  is the radius of the osculating circle at  $\Gamma(s)$ , with the same sign as the curvature, and where  $d\theta$  is the angle formed by the normals to the curve at  $\Gamma(s)$  and  $\Gamma(s + ds)$ , as illustrated in Fig. 4a. We also note that  $r = \phi(x_{(s,r)})$  and is negative when  $x$  is inside  $\Gamma$ , and positive when  $x$  is outside  $\Gamma$ . We can show that:

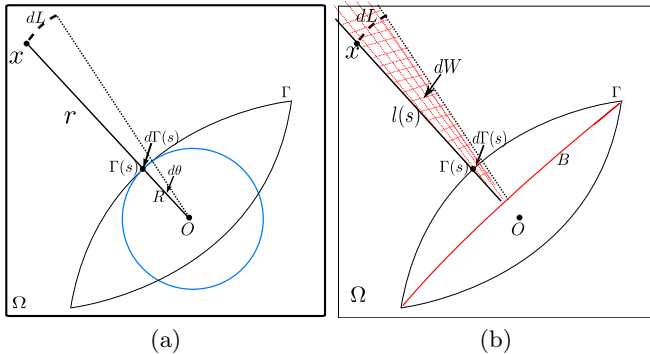
$$|1 - \phi(x)\kappa_{\Gamma(s)}| d\Gamma(s) = \left| 1 - \frac{r}{R} \right| d\Gamma(s) = \frac{|r - R|}{R} d\Gamma(s) = |r - R|d\theta = dL \quad (11)$$

where  $dL$  is the distance  $x_{(s,r)}$  will travel for an infinitesimal step  $ds$ , i.e. the length of the arc formed by the projection lines through  $\Gamma(s)$  and  $\Gamma(s + ds)$ , at a distance  $|r - R|$  from their intersection, as shown in Fig. 4a. Note that  $R < 0$  in this figure. Since in Eq.(9) we are integrating a function of  $x$  times  $dL(r)$  along the projection line segment bounded on one side by the skeleton, we are interested in the region formed by the skeleton, the boundary of  $\Omega$  and the projection lines going through  $\Gamma(s)$  and  $\Gamma(s + ds)$ , that is, the red dotted region  $dW$  shown in Fig. 4b. One can indeed show that, for any smooth function  $f$ :

$$\int_{dW(ds)} f(x) dx = \int_{r \in l(s)} f(x_{(s,r)}) dL(r) dr + o(\|\nabla_x f\|_\infty ds)$$

The above analysis shows that Eq.(9) can be written as an integral over this subregion  $dW$  when multiplied by an infinitesimal step  $d\Gamma(s)$ . We believe this is a more intuitive explanation of velocity projection, i.e. that the shape gradient at any point  $\Gamma(s)$  is the average limit of the level set gradient over the region whose projection points are between  $\Gamma(s)$  and  $\Gamma(s + ds)$ :

$$\nabla_{\Gamma} E(\Gamma)(s) = - \lim_{ds \rightarrow 0} \frac{1}{d\Gamma(s)} \int_{dW(ds)} \nabla_{\phi} F(\phi)(x) dx \mathbf{n}_{\Gamma(s)} \quad (12)$$



**Fig. 4.** Another look at velocity projection. (a) The osculating circle of  $\Gamma(s)$  is shown in blue, where  $O$  is the center of the osculating circle. (b) The skeleton of  $\Gamma$  is  $B$ , shown in red. The area of the dashed subregion  $dW$  is  $\int_{r \in l_s} dL(r) dr = \int_{r \in l_s} |1 - r \kappa_{\Gamma(s)}| dr d\Gamma(s)$ .

Although both Eq. (5) and Eq. (12) are the same theoretically, we recommend using the integral over  $dW$  for the following reason. To integrate along  $l(s)$ , we need to find the explicit range of  $r$ , whose estimation is not straightforward since it depends on the skeleton. We can however avoid the estimation of the skeleton if we integrate over regions  $W$ .

For any point  $x$  in  $\Omega$ , we can easily locate its projection point  $\Gamma(s_x) = x - \phi(x) \nabla_x \phi(x)$ . Suppose that  $\Gamma$  is discretized by points  $\Gamma(s_i), i = 1, \dots, N$ , and there is a  $W_i$  associated with each  $\Gamma(s_i)$ . Then Eq. (12) could be discretized as:

$$\nabla_{\Gamma} E(\Gamma)(s_i) = - \sum_{x \in W_i} \nabla_{\phi} F(\phi)(x) \mathbf{n}_{\Gamma(s_i)} \tag{13}$$

with its equivalent from Eq. (10) for direct level set evolutions, if  $x \in W_i$ :

$$\delta\phi(x) = \sum_{y \in W_i} \nabla_{\phi} F(\phi)(y) \tag{14}$$

which means that the correct level set evolution can be computed from the level set gradient  $\nabla_{\phi} F(\phi)$  very easily by just integrating over regions that share similar projection points.

However, since the projection  $s_x$  of a random point  $x \in \Omega$  is unlikely to be exactly one of the  $s_i$ , a point  $x$  will typically contribute to more than one  $W_i$ . Let us denote by  $h_i^x$  the weight that  $x$  contributes to  $W_i$ . We have the constraint that all contributions of a point sum up to 1 :  $\sum_i h_i^x = 1$ . Thus Eq. (13) is replaced by:

$$\nabla_{\Gamma} E(\Gamma)(s_i) = - \sum_{x \in \Omega} h_i^x \nabla_{\phi} F(\phi)(x) \mathbf{n}_{\Gamma(s_i)} \tag{15}$$

or, more practically, we obtain the level set evolution from Eq. (10):

$$\delta\phi(x) = \sum_i h_i^x \sum_{y \in \Omega} h_i^y \nabla_{\phi} F(\phi)(y). \tag{16}$$

Then the problem comes down to how to estimate the weight  $h_i^x$  that  $x$  contributes to  $W_i$  in a computationally effective manner. In practice, we assign  $h_i^x$  to a function of the distance from  $\Gamma(s_x)$  to  $\Gamma(s_i)$  and normalize accordingly. A better numerical implementation algorithm is also one of our future goals.

## 4 Implications for Common Level Set Gradients

As mentioned earlier, much work has been carried out under the variational level set method, without considering whether the level set gradient agrees with the shape gradient. In this section, we will discuss some energy models that depend on SDFs and their gradients. We will not discuss energy models that depend only on  $\Gamma$ , such as the Geodesic Active Contour [16] and the Chan-Vese models [6], since our aim is to compare shape gradients with level set gradients.



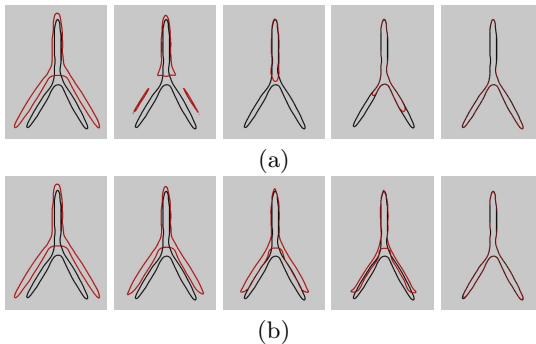
We first consider the  $L^2$  distance between two SDFs and its gradient:

$$F(\phi) = \|\phi - \phi_T\|_{L^2}^2, \quad \nabla_\phi F(\phi) = 2(\phi - \phi_T) \tag{17}$$

Here  $\phi_T$  is a target SDF. This energy is important and has many applications in shape analysis, morphing and shape prior image segmentation [17,18]. We are not aware of any work on computing the corresponding  $E(\Gamma)$  or its shape gradient  $\nabla_\Gamma E(\Gamma)$ . Charpiat et al. [19] showed how to calculate the shape gradient directly with the  $W^{1,2}$  norm but under a smooth approximation of infima. We can easily compute the shape gradient with our velocity projection step.

If instead, as in other works, we let  $\phi$  evolve with the level set gradient, and rebuild it regularly from its 0-level to maintain its SDF property, we notice the following effect. Curve segments of  $\Gamma$  that lie inside the region delimited by  $\Gamma_T$  expand, while segments of  $\Gamma$  that lie outside of  $\Gamma_T$  shrink. This immediately implies that if  $\Gamma$  lies completely outside of  $\Gamma_T$ , then the evolution process will shrink  $\Gamma$  until it disappears no matter how close they are. This phenomenon is illustrated in Fig. 4a where we are trying to evolve the red circle  $\Gamma$  to the black circle  $\Gamma_T$ . With the velocity projection approach, we can calculate the shape gradient and deform  $\Gamma$  accordingly. Fig. 4b illustrates the deformation process under the correct shape gradient, which naturally morphs the initial curve to  $\Gamma_T$ .

Fig. 5 illustrates the evolution of two overlapped shapes using the same  $L^2$  distance with and without velocity projection. Fig. 5a illustrates the traditional evolution without velocity projection. As we can see, the parts that are outside  $\Gamma_T$  will shrink while the parts that are inside will expand. Fig. 5b illustrates the correct deformation with velocity projection. As we can see, the deformation is more meaningful, leading to much better point correspondences. However, this model suffers from the drawback that the energy and thus the gradient depend on the domain  $\Omega$ . That is, by fixing  $\Gamma$  and  $\Gamma_T$  and changing  $\Omega$  alone, we will get different energies, gradients and thus different deformation processes.



**Fig. 5.** Curve evolution of the red curve  $\Gamma$ , where the black curve  $\Gamma_T$  is the target. The cost function is the  $L^2$  distance between the two SDFs. (a) standard variational level set method, (b) our result.

As a second example, consider the following energy model:

$$F(\phi) = \int_{\Omega} (\phi - \phi_T)^2 H(-\phi) dx \tag{18}$$

which is the integration of the  $L^2$  distance between  $\phi$  and  $\phi_T$  inside  $\Gamma$ . Here  $H$  is the Heaviside function. This energy model was studied by Rousson and Paragios [7]. It can be shown that the level set gradient is:

$$\nabla_{\phi} F(\phi) = 2(\phi - \phi_T)H(-\phi) - (\phi - \phi_T)^2 \delta(\phi) \tag{19}$$

In this case, the velocity projection step is necessary to calculate the correct shape gradient. The evolution process of the correct shape gradient is illustrated in Fig. 6a. Since the integration is only inside  $\Gamma$ , it is not appropriate if  $\Gamma$  lies outside  $\Gamma_T$ . To improve the evolution, we study the following symmetric term:

$$F(\phi) = \int_{\Omega} (\phi - \phi_T)^2 H(-\phi_T) dx \tag{20}$$

which is the integration of the  $L^2$  distance between  $\phi$  and  $\phi_T$  inside  $\Gamma_T$ . This energy was studied by Cremers and Soatto [8]. The level set gradient is:

$$\nabla_{\phi} F(\phi) = 2(\phi - \phi_T)H(-\phi_T) \tag{21}$$

It is only defined within  $\Gamma_T$  and therefore if we try to make  $\Gamma$  evolve to  $\Gamma_T$  under this gradient alone, most likely it won't move at all! We can again calculate its shape gradient (see Fig. 6b). This evolution does not correctly draw  $\Gamma$  to  $\Gamma_T$ . The reason is that the level set gradient is non-0 inside  $\Gamma_T$  and all the projection points inside  $\Gamma_T$  fall only on the blue curve segment of  $\Gamma$  in Fig. 6c. Therefore, the fastest way to minimize this energy, or the gradient, is to only move the blue curve segment to  $\Gamma_T$ . The dissimilarity measure between  $\Gamma$  and  $\Gamma_T$  can be made symmetric by combining both Eq. (18) and Eq. (20):

$$F(\phi) = \int_{\Omega} (\phi - \phi_T)^2 H(-\phi_T) dx + \int_{\Omega} (\phi - \phi_T)^2 H(-\phi) dx \tag{22}$$

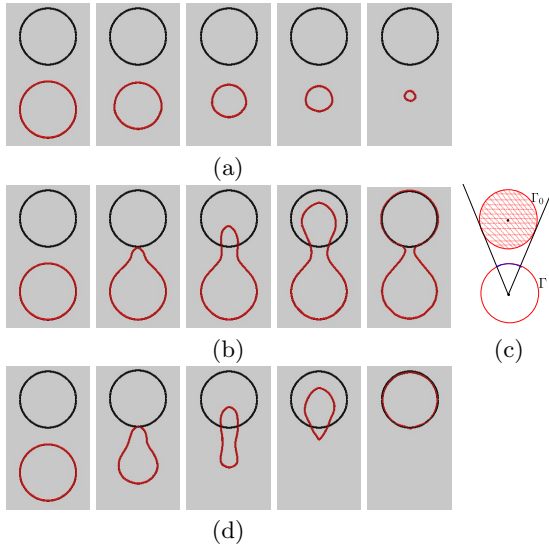
The evolution of this symmetric dissimilarity measure is shown in Fig. 6d. As we can see, it correctly evolves  $\Gamma$  to  $\Gamma_T$ . We note here that the traditional variational gradient of this symmetric measure could also be used to evolve  $\Gamma$ . However, since the level set gradient Eq. (21) is not defined on curve segments of  $\Gamma$  that lie outside  $\Gamma_T$ , only the gradient Eq. (19) would play a role in the evolution and it would not correctly draw  $\Gamma$  to  $\Gamma_T$ .

As a final example, we consider the following energy model [20]:

$$E(\Gamma) = \int_{\Gamma_T} \phi^2 ds = \int_{\Omega} \phi^2 \delta(\phi_T) dx = F(\phi) \tag{23}$$

This also defines a dissimilarity between  $\Gamma$  and  $\Gamma_T$ . The level set gradient is:

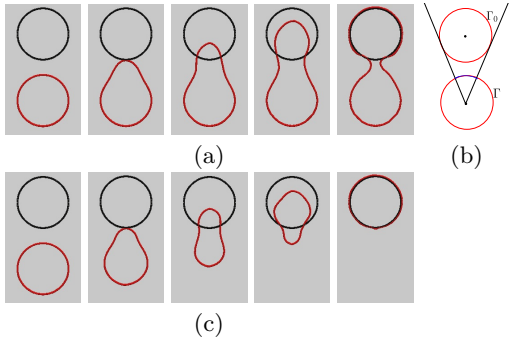
$$\nabla_{\phi} F(\phi) = 2\phi \delta(\phi_T) \tag{24}$$



**Fig. 6.** The evolution from the red circle  $\Gamma$  to the black circle  $\Gamma_T$  under the correct shape gradient of the three different energy models. (a) energy model Eq. (18), (b) energy model Eq. (20), (c) the blue curve segment is the region that  $H(-\phi_T)$  projects onto for (b), (d) energy model Eq. (22).

which is defined only along  $\Gamma_T$  and is zero everywhere else. Therefore, it is also problematic for the traditional variational level set method. However, if we apply the velocity projection approach we can calculate the true deformation field:

$$\delta\phi(x) = -2 \sum_{y \in (l(s_x) \cap \Gamma_T)} |1 - \phi(y)\kappa(s_x)| \phi(y) \tag{25}$$



**Fig. 7.** The evolution from the red circle  $\Gamma$  to the black circle  $\Gamma_T$  under the shape gradient of two different energy models. (a) energy model Eq. (23), (b) the blue curve segment is the region that  $\delta(\phi_T)$  projects onto, (c) energy model Eq. (26).

where  $l(s_x)$  is the projection line going through  $x$ . The deformation process is illustrated in Fig. 7a. However, as we can see, this evolution also does not correctly draw  $\Gamma$  to  $\Gamma_T$ . The reason is the same as in Fig. 6c.

To improve the evolution process, we can add a second symmetric term:

$$F(\phi) = \int_{\Gamma_T} \phi^2 ds + \int_{\Gamma} \phi_T^2 ds \quad (26)$$

The evolution of this symmetric energy under the correct shape gradient is illustrated in Fig. 7c and it successfully draws  $\Gamma$  to  $\Gamma_T$ .

## 5 Discussion and Conclusions

The experiments in Section 4 show that the limitations of traditional variational level set formulations can be fixed with our velocity projection step. In this work, we used shape morphing as a motivating application since it is closely related to other computer vision problems. In future work, we plan to apply our method to shape-prior image segmentation and statistical shape analysis. Fig. 5 shows the evolution of tubular structures under the traditional level set gradient can be problematic and our corrected shape gradient can handle these cases nicely.

We should point out that the geometric  $L^2(\mathbb{S}^1 \rightarrow \mathbb{R})$  inner product has been shown to suffer from serious drawbacks as a metric on the manifold of shapes [21,22]. Specifically, the  $L^2$  geodesic distance between two shapes is 0. Since we consider gradient descents only, the  $L^2$  inner product will pose no theoretical problem. However, we would like to investigate other inner products such as the  $H^1(\mathbb{S}^1 \rightarrow \mathbb{R})$  inner product [14,15]. We are also investigating energy models other than  $F(\phi)$ , such as  $F(\Gamma, \phi)$ , as well as extending our framework to 3D.

## References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. *International Journal of Computer Vision* 1, 321–331 (1988)
2. Osher, S., Sethian, J.: Fronts propagating with curvature-dependent speed- Algorithms based on Hamilton-Jacobi formulations. *J. of Comp. Physics* 79 (1988)
3. Osher, S., Fedkiw, R.: *Level set methods and dynamic implicit surfaces* (2002)
4. Aubert, G., Barlaud, M., Faugeras, O., Jehan-Besson, S.: Image segmentation using active contours: Calculus of variations or shape gradients? *SIAM Journal of Applied Mathematics* 63 (2003)
5. Zhao, H., Chan, T., Merriman, B., Osher, S.: A Variational Level Set Approach to Multiphase Motion. *Journal of Computational Physics* 127, 179–195 (1996)
6. Chan, T., Vese, L.: Active contours without edges. *IEEE TIP* 10, 266–277 (2001)
7. Rousson, M., Paragios, N.: Shape Priors for Level Set Representations. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*. LNCS, vol. 2351, pp. 78–92. Springer, Heidelberg (2002)
8. Cremers, D., Soatto, S.: A pseudo-distance for shape priors in level set segmentation. In: *IEEE workshop on variational, geometric and level set methods* (2003)
9. Gomes, J., Faugeras, O.: Reconciling distance functions and level sets. *Journal of Visual Communication and Image Representation* 11, 209–223 (2000)

10. Adalsteinsson, D., Sethian, J.: The fast construction of extension velocities in level set methods. *Journal of Computational Physics* 148, 2–22 (1999)
11. Solem, J., Overgaard, N.: A geometric formulation of gradient descent for variational problems with moving surfaces. In: Kimmel, R., Sochen, N.A., Weickert, J. (eds.) *Scale-Space 2005*. LNCS, vol. 3459, pp. 419–430. Springer, Heidelberg (2005)
12. Peng, D., Merriman, B., Osher, S., Zhao, H., Kang, M.: A PDE-Based Fast Local Level Set Method. *Journal of Computational Physics* 155, 410–438 (1999)
13. Li, C., Xu, C., Gui, C., Fox, M.: Level set evolution without re-initialization: A new variational formulation. In: *CVPR*, pp. 430–436 (2005)
14. Charpiat, G., Maurel, P., Pons, J., Keriven, R., Faugeras, O.: Generalized gradients: Priors on minimization flows. *IJCV* 73, 325–344 (2007)
15. Sundaramoorthi, G., Yezzi, A., Mennucci, A.: Sobolev active contours. *International Journal of Computer Vision* 73, 345–366 (2007)
16. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. *IJCV* 22 (1997)
17. Kim, J., Cetin, M., Willsky, A.: Nonparametric shape priors for active contour-based image segmentation. *Signal Processing* 87, 3021–3044 (2007)
18. Cremers, D., Sochen, N., Schnorr, C.: Towards Recognition-Based Variational Segmentation Using Shape Priors and Dynamic Labeling. In: *SSVM* (2003)
19. Charpiat, G., Faugeras, O., Keriven, R.: Approximations of shape metrics and application to shape warping and empirical shape statistics. *Foundations of Computational Mathematics* 5, 1–58 (2005)
20. Gui, L., Thiran, J., Paragios, N.: Cooperative object segmentation and behavior inference in image sequences. *IJCV* 84, 146–162 (2009)
21. Michor, P., Mumford, D.: Riemannian geometries on spaces of plane curves. *Arxiv preprint math/0312384* (2003)
22. Yezzi, A., Mennucci, A.: Conformal metrics and true “gradient flows” for curves. In: *ICCV*, vol. 1 (2005)

## Appendix

Proof of Eq. (5):  $(\phi + \delta\phi)(x) = \phi_{\Gamma'}(x)$  implies, using the signed distance  $d$ :

$$\delta\phi(x) = \phi_{\Gamma'}(x) - \phi(x) = d(x, \Gamma') - d(x, \Gamma). \quad (27)$$

Since the part of an infinitesimal deformation that is tangent to  $\Gamma$  has no effect on the shape (just reparameterizes), we only keep the part of the deformation that is normal to  $\Gamma$ , and the following redefinition of  $\Gamma'$  describes the same shape (as a set of points):  $\Gamma'(s) := \Gamma(s) + (\delta\Gamma(s) \cdot \mathbf{n}_{\Gamma(s)})\mathbf{n}_{\Gamma(s_x)}$ .

$$\begin{aligned} x - \Gamma'(s_x) &= x - \Gamma(s_x) - (\delta\Gamma(s_x) \cdot \mathbf{n}_{\Gamma(s_x)})\mathbf{n}_{\Gamma(s_x)} \\ (x - \Gamma'(s_x)) \cdot \mathbf{n}_{\Gamma(s_x)} &= (x - \Gamma(s_x)) \cdot \mathbf{n}_{\Gamma(s_x)} - (\delta\Gamma(s_x) \cdot \mathbf{n}_{\Gamma(s_x)})(\mathbf{n}_{\Gamma(s_x)} \cdot \mathbf{n}_{\Gamma(s_x)}) \\ d(x, \Gamma') &= d(x, \Gamma) - \delta\Gamma(s_x) \cdot \mathbf{n}_{\Gamma(s_x)} \end{aligned}$$

since the projection of any point  $x \in \Omega$  on the closed subset  $\Gamma$  is necessarily orthogonal to its boundary  $\Gamma$ . Hence with Eq. (27) one has  $\delta\phi(x) = -\delta\Gamma(s_x) \cdot \mathbf{n}_{\Gamma(s_x)}$ .

# A Close-Form Iterative Algorithm for Depth Inferring from a Single Image

Yang Cao, Yan Xia, and Zengfu Wang

Automation Department, University of Science and Technology of China,  
Jinzhai Road 96, Hefei, China

forrest@ustc.edu.cn, xiayan@mail.ustc.edu.cn, zfwang@ustc.edu.cn

**Abstract.** Inferring depth from a single image is a difficult task in computer vision, which needs to utilize adequate monocular cues contained in the image. Inspired by Saxena et al's work, this paper presents a close-form iterative algorithm to process multi-scale image segmentation and depth inferring alternately, which can significantly improve segmentation and depth estimate results. First, an EM-based algorithm is applied to obtain an initial multi-scale image segmentation result. Then, the multi-scale Markov random field (MRF) model, trained by supervised learning, is used to infer both depths and the relations between depths at different image regions. Next, a graph-based region merging algorithm is applied to merge the segmentations at the larger scales by incorporating the inferred depths. At the last, the refined multi-scale image segmentations are used as input of MRF model and the depth are re-inferred. The above processes are iteratively continued until the expected results are achieved. Since there are no changes on the segmentations at the finest scale in the iterative process, it still can capture the detailed 3D structure. Meanwhile, the refined segmentations at the other scales will help obtain more global structure information in the image. The contrastive experimental results verify the validity of our method that it can infer quantitatively better depth estimations for 62.7% of 134 images downloaded from the Saxena's database. Our method can also improve the image segmentation results in the sense of scene interpretation. Moreover, the paper extends the method to estimate the depth of the scene with fore-objects.

**Keywords:** Depth inferring, monocular cues, image segmentation, Markov random field, scene reconstruction.

## 1 Introduction

Inferring 3D scene structure from a single image is an extremely challenging topic in computer vision, since it is an ill-posed problem in a mathematical sense and we can never know if the image is a picture of a painting or if it is a picture of an actual 3D environment. However, people have no difficulty to infer the scene structure from a single image. Here people utilize monocular depth cues to infer 3D information, which include some physical phenomenon

and object characteristics, such as lighting and shading, perspective, occlusion, texture gradient and so on.

In recent works, researchers exploited some of these cues to obtain some 3D information from a single image. Saxena et al. [1,2,3,4,5] presented a Markov random field model for inferring depths from multi-scale monocular image features and applied the monocular depth perception to drive a remote-controlled car autonomously. Hoiem et al. [6,7,8] used texture and perspective cues to build pop-up models under a strong assumption that the scene consists of ground/horizontal planes and vertical walls (and possibly sky). Based on this, Hoiem et al. [9] also presented a closed form framework to integrate surface orientations, occlusion boundaries and objective identifications to develop a 3D scene understanding system. But the methods cannot be applied to the many scenes that are not made up only of vertical surfaces standing on a horizontal floor [10], such as mountains, trees, rooftops and so on.

In this paper, the goal is to propose a close-form iterative algorithm for improving the accuracy of depth inferring. In Hoiem et al's and Saxena et al's work, the depths of nature scene are approximately inferred from an over-segmentation of the image under the assumption that the 3D scene is made up of a number of small planes. This implies that image segmentation and depth inferring are inter-correlated. The image segmentations can help inferring the relations between the depths of different image regions. On the other hand, the depths can also be used as an additional attribute to improve segmentation results. Our algorithm utilizes this inter-correlated property and processes image segmentation and depth inferring alternately.

As mentioned in Saxena et al's work, local image features are insufficient to estimate the depth and multi-scale image features have to be used to capture more global properties. So we apply an EM-based multi-scale image segmentation algorithm to obtain the initial segmentation results. The image feature vectors extracted from the multi-scale segmentations are used to infer the different depth of each pixel in the image. The inferred depths are fed back and integrated with image segmentation into a cognitive loop. It is particularly noted that the depth inferring is regarding the segmentation regions at the finest scale, while the region merging is acting on the regions at the larger scales. This method will not decrease the number of the patches made up of 3D scene structure and can capture the rich detailed 3D scene structure. At the same time, the refined segmentations at the larger scales can offer more global structure information in multiple spatial scales which can be used to improve the accuracy of depth inferring. The above processes are iteratively continued until the expected results are achieved.

By using this close-form iterative framework, our algorithm can significantly improve the depth estimation results. Compared with the exiting methods, our algorithm can provide sharper depthmaps for 62.7% of 134 test images. The 3D flythrough reconstruct results using our algorithm are also a bit more visually pleasing. In additional, our method can improve the image segmentation results in the sense of scene interpretation.

Furthermore, we also consider the problem of depth inferring for scene with fore-objects. Under the assumption that the fore-objects lie vertical on the ground, the fore-objects regions are extracted from the image and the depth inferring of these regions are processed solely. After the other regions have also been processed, the depth estimations are incorporated together.

The remainder of this paper is organized as follows. Related works are reviewed in section 2. The overview of the proposed algorithm is introduced in section 3. The close-form iterative algorithm is described in section 4. Experimental results are shown in section 5. The depth inferring method for scene with fore-object is explained in section 6, before concluding in section 7.

## 2 Related Works

In some specific settings, monocular cues have been applied to perform the tasks of depth inferring from a single image. A number of researchers have studied the corresponding problems and proposed some effective methods including shape from texture (SFT) [11,12], shape from shading (SFS) [13,14] and tour into picture (TIP) [15]. Different from the geometric methods relied on feature matching and triangulation, such as stereo vision [16] and shape from motion [17], these methods use the cues contained in image to obtain rich 3D information. However, these methods often ignore the additional useful cues and enforce hard assumption that the scene structure is simple and uniform, thus they can only be applied in limited environment. For example, the TIP method can only be used in fully structured environment.

Recently, great progresses have been made in applying monocular cues to obtain 3D information. Based on the assumption that the environment is made of a ground-vertical structure, Delage et al. [18] and Hoiem et al. [6,7], built a simple pop-up 3D model from an image by classifying the image into horizontal/ground and vertical regions (also possibly sky). Delage considered indoor images, while Hoiem considered outdoor scenes. Based on these concepts, Hoiem et al. [10] and Sudderth et al. [19] integrated learning-based object recognition with 3D scene reconstruction; Hedau et al. [8] presented an algorithm to recover the spatial layout of cluttered room. Saxena et al. [1,2,4,5] presented an algorithm for inferring depth from monocular image cues. This algorithm was also successfully applied for improving the performance of stereovision [3] and autonomous navigation of remote-controlled car [20]. Heitz et al. [21] developed cascaded classification models (CCM) that combined a set of related subtasks of scene categorization, object detection and 3d reconstruction and these tasks can be solved in its own level and help each other. Hoiem et al. [9] regarded surface orientations, occlusion boundaries and objective identifications as intrinsic images and presented a closed form framework for interfacing scene analysis processes.

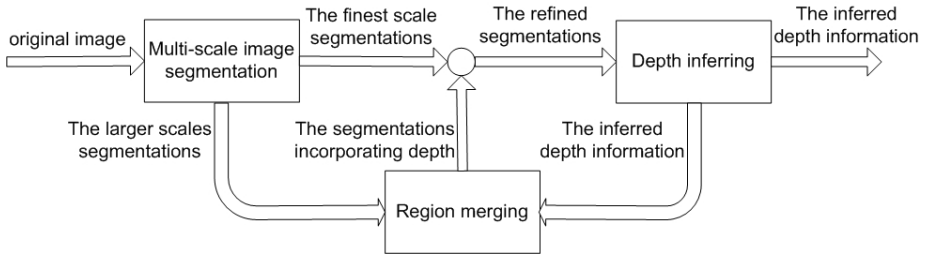
Our work seems like Heitz et al's and Hoiem et al's works for integrating the tasks of image segmentation and depth inferring. However, their works have strong leaning towards image understanding rather than depth inferring, and their algorithms contain many steps include object detection, region labeling and



so on. Moreover, their algorithms are based on iterative training which requires the knowledge of the implementation of each step, while our algorithm does not need retraining and is more flexible to be used in some specific applications such as robot navigation.

### 3 Overview of Our Algorithm

The overview of our proposed algorithm is illustrated in Fig. 1. There are three main modules, image segmentation, depth inferring, and region merging. Our input data are the multi-scale image segmentations, obtained by an EM-based algorithm at different scales. From these multi-scale segmentations, image feature vectors are first extracted through a template. Subsequently, a multi-scale Markov random field, trained by supervised learning, is used to model the relations between image feature vectors and the different depths of image regions at the finest scale. Then the inferred depths are fed back to incorporate the larger-scale image segmentations that are closed in 3D structure. Combined with the initial segmentation at the finest scale, the refined multi-scale segmentation results are obtained. The above processes are iteratively continued until the expected depth inferring results are achieved.



**Fig. 1.** Overview of our algorithm integrating depth inferring with image segmentation

The three modules are integrated in a cognitive loop. For each image, the region merging module receives initial segmentations and depth information from the other two modules and feeds back the refined multi-scale segmentations. Thus, the modules exchange information that helps compensate for their individual disadvantages and improves overall system performance. The pipeline of our algorithm is detailed in the following sections.

## 4 The Framework of Our Algorithm

### 4.1 Multi-scale Image Segmentation

Same as Hoiem et al's and Saxena et al's works, our algorithm also begins by segmenting the image into many such small planar surfaces. In order to capture

the depth cues directly from the local structure of the brightness pattern of a single monocular image, we use an expectation-maximization image segmentation algorithm [22,23] to obtain the initial segmentation results. The algorithm can offer an effective solution to bridge the gap from the low-level image features to surface reconstruction. Due to the extent of the inner-workings of the algorithm, we refrain from explaining in detail of the well-known algorithm, but limit the introduction to our specific application of the algorithm.

Creating multi-scale segmentation of an image involves three steps. (1) Select an appropriate scale for each pixel, and then extract color, texture and position features for that pixel at the selected scale. (2) Group pixels into regions by modeling the distribution of pixel features with a mixture of Gaussian using Expectation-Maximization. (3) Repeat the above two steps at multiple spatial scales.

In this image segmentation algorithm, the pixels are represented by the descriptor consists of eight values: three for color, three for texture and two for position. The three color components are the coordinates of Lab color space, which is approximately perceptually uniform and has the distances to be meaningful. The three texture components are polarity, anisotropy and contrast of each pixel, computed at the selected scale. The anisotropy and polarity are each modulated by the contrast since they are meaningless in regions of low contrast. The position of the pixel in the image, which can describe the spatial distribution, is also included in the feature vector.

Then, the Expectation-Maximization (EM) algorithm is applied to segment the pixels into patches. Since an image can be regarded as points in an eight-dimensional feature space after the process of feature extraction, the segmentation problem are transformed into dividing these points into groups. So the EM algorithm is actually used to determine the maximum likelihood parameters by assuming  $K$  Gaussian mixture model in the feature space. In order to avoid under-segmentations, we choose a rather large value of  $K$ , where  $K=256$  for a  $1024 \times 768$  size image in our experiment.

In order to capture more global structure properties from image, the segmentation algorithm are applied at three different scales (image resolutions, which are 1x, 3x and 9x of the original one in our experiments). The segmentations at the larger two scales are to be replaced by the refined ones after the region merging. An example result is shown as Fig 2.



**Fig. 2.** (a) Original image, (b)-(d) multi-scale image segmentation results

### 4.2 Depth Inferring

**Feature Vector.** In our algorithm, we choose the same features with Saxena et al’s. There are two types of features: absolute features and relative features, which are used to estimate the absolute depth and relative depths respectively. As described in [2], a 17 dimensional template consists of 9 Laws’masks, 2 color channels and 6 texture gradients are used to compute summary statistics for a patch  $i$  at scale  $s$  in the image  $I$ . For the absolute depth feature, the outputs are incorporated to compute the sum absolute energy and sum squared energy. After including features from itself and its 4 neighbors at 3 scales and its 4 location features, the absolute feature vector  $x$  is  $19 \times 34 = 646$  dimensional. For the relative depth features, a 10-bin histogram of each of the template output is computed, giving us a total of 170 features  $y_{is}$  for each patch  $i$ . Then the 170 dimensional relative depth features vector  $y_{ijs}$  for two neighboring patches  $i$  and  $j$  at scale  $s$  are computed as  $y_{ijs} = y_{is} - y_{js}$ .

**Multi-scale Markov random model.** The monocular depth cues of a particular patch are not only contained in this patch, but also can be captured from the relations between the patches which are adjacent at multiple spatial scales. Similar to Saxena et al’s work [2,5], a hierarchical multi-scale Markov Random Field (MRF) is used to model the relationship between the depth of a patch and the depths of its neighboring patches. The model is formulated as,

$$P(d|X; \theta, \sigma) = \frac{1}{Z} \prod_i^K \prod_{p_i}^{P_i=1} f_1(d_{i,p_i} | X_{i,p_i}, \theta_r, \sigma_{1r}) \prod_{s=1}^3 \prod_i^K \prod_{j \in N_s} f_2(d_i(s), d_j(s) | y_{ijs}, \sigma_{2r}). \tag{1}$$

where  $Z$  is the normalization constant for the model;  $K$  is the total number of patches in the image (at the lowest scale); with a total of  $P_t$  points in the patch  $i$ ,  $X_{i,p_i} = \{ \in R^{646}, p_i = 1, 2, 3 \dots P_i \}$  is the absolute depth feature vector for the point  $p_i$  in the patch  $i$ ;  $s = \{1, 2, 3\}$  is the 3 scales of image;  $N_s(i)$  are the 4 neighbors of patch  $i$  at scale  $s$ ;  $\theta_r, \sigma_{1r}, \sigma_{2r}$  are the parameters of the model. The model consists of two terms,  $f_1(\cdot)$  and  $f_2(\cdot)$ . The first term  $f_1(\cdot)$  captures the relations between the depth  $d_{i,p_i}$  and the absolute feature  $X_{i,p_i}$  and it is formulated as,

$$f_1(\cdot) = \exp(-|d_{i,p_i}(1) - x_{i,p_i}^T \theta_r | \sigma_{1r}). \tag{2}$$

The parameter  $\sigma_{1r}$  is modeled as a linear function of the features, which is  $\sigma_{1r} = u_r^T x_{i,p_i}$ . The second term  $f_2(\cdot)$  captures the relations between the depths of patches which are adjacent at multiple spatial scales and it is formulated as,

$$f_2(\cdot) = \exp(-|d_i(s) - d_j(s) | \sigma_{2r}). \tag{3}$$

In our algorithm, there are two constraints on the depths of patch  $i$  at the scale  $s$ . The first one is that the depths of patch  $i$  are the average of the depths of all of points in patch  $i$ .

$$d_i(s) = 1/P_i \sum_{p_i=1}^{P_i} d_{i,p_i}(s). \tag{4}$$

The second one is that the depths at a higher scale are the average of the depths at the lower scale.

$$d_i(s + 1) = 1/5 \sum_{j \in N_s(i) \cup i} d_j(s) . \tag{5}$$

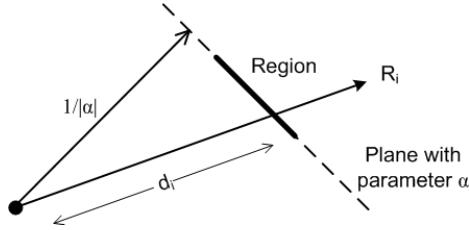
Similar to the parameter  $\sigma_{1r}, \sigma_{2r}$  is modeled as  $\sigma_{2r} = v_{rs}^T y_{ijs}$ . In detail, different parameters  $(\theta_r, u_r, v_r)$  are used for each row  $r$  in the image to learn the different statistical properties of different rows of image. Since the location features are also included in image segmentation and feature extraction, it can improve to detect some specific regions, such as sky and ground. For example, a blue region might represent sky if it is in upper part of image, and a green region might be more likely to be ground if in the lower part of the image.

**Parameter Learning and MAP Inference.** As described in [2], an approximate parameter learning of the model is made by using Multi-Conditional Learning (MCL). With  $u_r \geq 0$  and  $v_{rs} \geq 0$ , the model parameters are estimated by solving a Linear Program (LP). After learning the parameters, the depth inferring problem is transformed into the MAP inference problem by maximizing (1) in terms of  $d$ . It can be seen that the first term in (1) models depth as an exponential function of multi-scale features of the points in the single patch  $i$ . The second term places a constraint that depends on the multi-scale relative features  $y_{ijs}$  on the depths, which plays a role to improve the accuracy of initial depth estimates. The MAP inference of the depth  $d_i$  can also be performed by solving a LP.

### 4.3 Region Merging

Region merging is the core part of our algorithm. As shown in Fig. 1, the inputs of region merging module are the inferred depth and the initial image segmentation results, and the outputs are the refined segmentations at the two larger scales. With this module, our algorithm can capture the strong interactions between the depths of patches which are not immediate neighbors. For example, consider the patches that lie on a large building, which are to be at similar depths. However, some adjacent patches are difficult to recognize as parts of the same object, since there are discontinuities in feature space (such as a window on the wall of a building). When the depth information is fed back, the adjacent patches tend to be incorporated and the discontinuities are eliminated. Then the depths of the patches will be highly correlated according to the MRF model.

As described above, each segmentation region represents a coherent region in the scene with all the pixels having similar properties. Thus, the 3D scene model is assumed to be made of a set of small planes. For ease of description, the basic unit of representation in the region merging module will be these small planes in the world. The relations between depths and the planar parameters are described as Fig 3. The planar surface on which a segmentation region lies is represented by using a set of plane parameters  $\alpha \in R^3$ , as described in [5]. The



**Fig. 3.** Illustration of the relations between plane parameter  $\alpha$  and the depth  $d$  of the point  $i$  (Cited from [5])

camera viewpoint is a two-parameter and is assumed to be constant. The value  $1/|\alpha|$  is the distance from the camera center to the closest point on the plane, and the normal vector  $\hat{\alpha} = \alpha/|\alpha|$  gives the orientation of the plane.  $R_i$  is the unit vector from the camera center to a point  $i$  lying on a plane with parameters  $\alpha$ . Then the planar parameter  $\alpha$  can be computed by least square fitting according to  $R_i$  and the estimated depth  $d_i$  at the corresponding plane.

Then, the relations between two adjacent regions are weighted by the angle between the two planes on which the two regions lie. The weighted function  $W(i, j)$  is defined as,

$$W(i, j) = \begin{cases} \theta_{ij} & \text{if the region } i \text{ is adjacent to region } j \\ \infty & \text{if the region } i \text{ is not adjacent to region } j \end{cases} \quad (6)$$

$$\theta_{ij} = |\alpha_i/|\alpha_i| - \alpha_j/|\alpha_j|| \quad (7)$$

where  $\theta_{ij}$  is the angle between two planes  $i$  and  $j$ . A graph-based segmentation algorithm is used to realize region merging. The image is abstracted into an undirected weighted graph  $G(V, E)$ .  $V$  is the vertex set of  $G$  with its elements  $V_i$  representing the regions.  $E$  is the edge set of  $G$  with its elements  $E_{i,j}$  representing the relations between two vertexes  $V_i$  and  $V_j$ . Based on the obtained graph  $G(V, E)$ , the region merging algorithm is performed as described in [24]. An example result is shown as Fig.4.



**Fig. 4.** (a) Image segmentation region at the largest region, (b) Initial depth reconstruction result,(c) Region merging result. (Best viewed in color)

## 5 Experiments

In order to verify the validity of our approach, we performed contrastive experiments that compare our algorithm with Saxena et al’s [25], and Hoiem et al’s work [10]. We downloaded 534 images+depthmaps from Saxena’s homepage and used 400 for training model. The rest 134 images are used for quantitative comparison and the other 150 internet images are used for qualitative comparison.

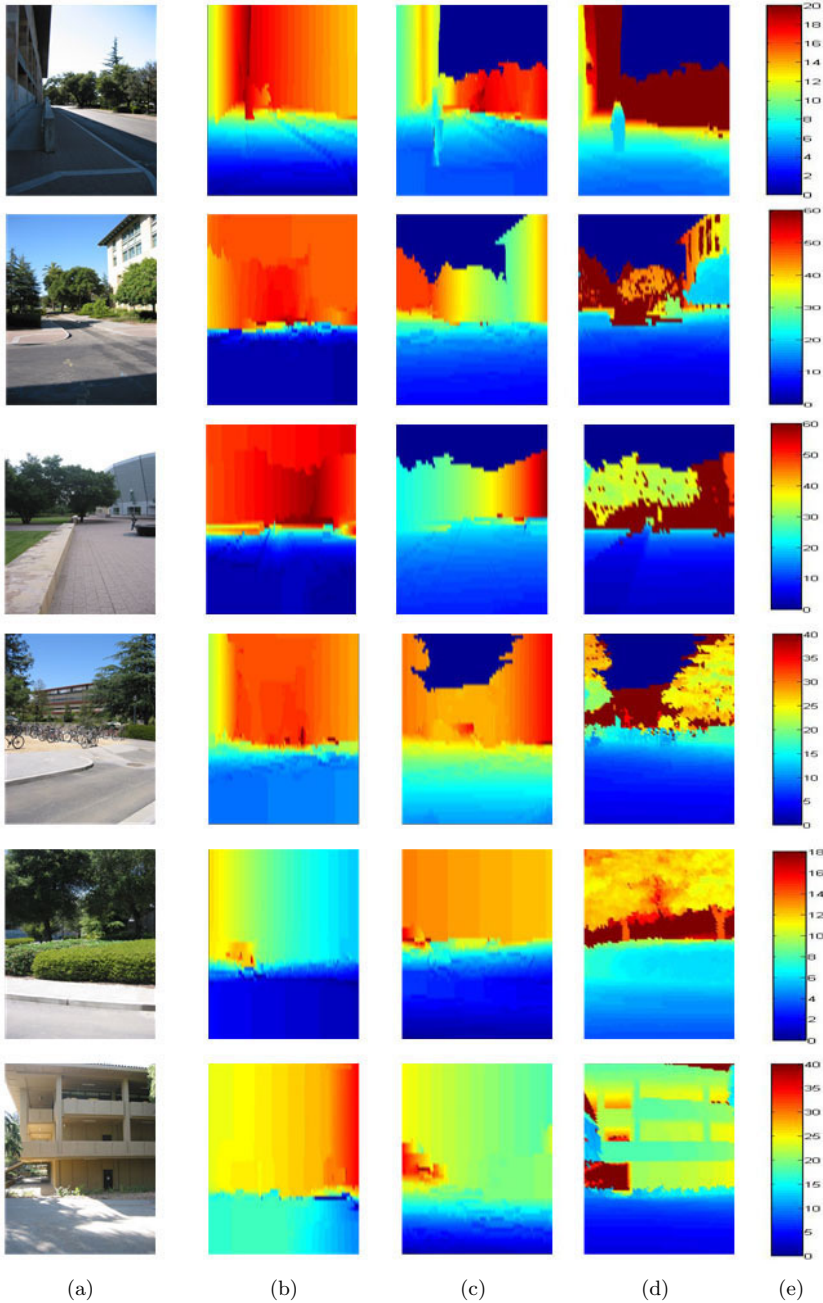
We use relative depth error  $|d - \hat{d}|/d$  as the performance metric to decide which algorithm is quantitative better. We also perform a further qualitative comparison experiment that we ask a person to compare the three 3D fly through results, and decide which algorithm is qualitative better. The quantitative and the qualitative comparison results are shown in the Table 1. Since Hoiem et al’s work is leaning towards surface reconstruction rather than depth inferring, the average relative depth errors are rather large, but the scene reconstruction results are more visual pleasing. Compared with Saxena et al’s work, our method gives better relative depth accuracy for 62.7% of 134 images. Our algorithm also outputs visually better model in 35% of the cases, while Saxena et al’s method outputs better model in 21% cases and Hoiem et al’s work outputs better model in 35% cases (the rest cases are hard to decide).

**Table 1.** The quantitative and qualitative comparison results

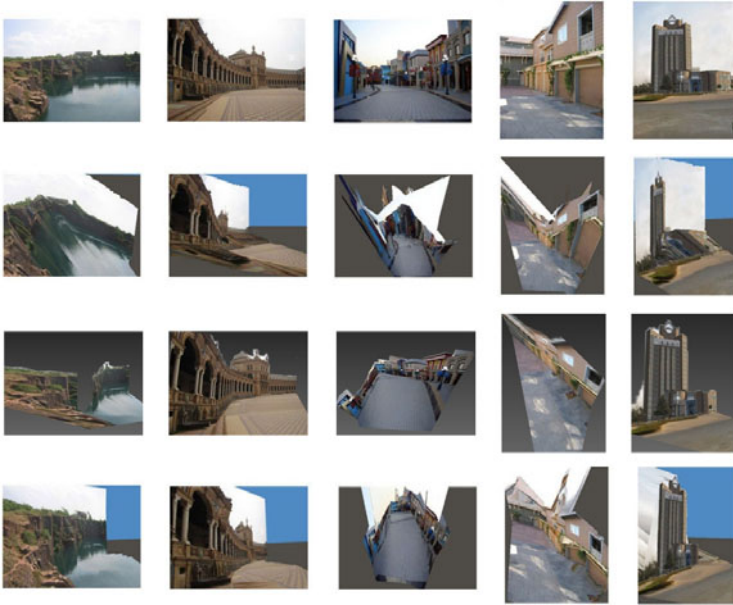
Algorithm	Quantitative better	Average relative depth error	Qualitative better
Hoiem et al’s	0.7%	4.055	35%
Saxena et al’s	36.6%	0.400	21%
Our	62.7%	0.312	35%

The inferring depthmap compared with Saxena et al’s work and ground truth are shown in Fig 5 and the typical scene reconstruction results are shown in Fig 6. As seen in the 3<sup>rd</sup> image at the 2<sup>nd</sup> row in Fig 5 and the 2<sup>nd</sup> image at the 4<sup>th</sup> row in Fig 6, the details of the distant region in image are arbitrarily to be reconstructed as a uniform one due to using depth information. Although the region merging is only acted on the regions at the larger scales in order to improve this, this situation still happens sometimes. On the whole, nevertheless, using the close form iterative framework yields better reconstruct results than before.

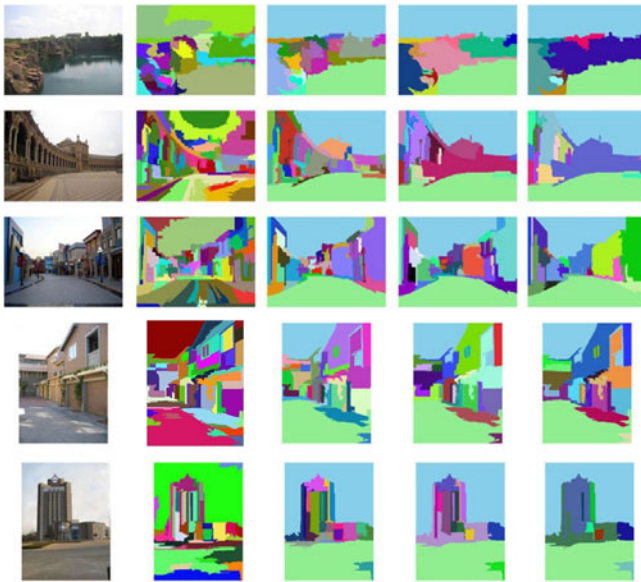
As a byproduct of our algorithm, the image segmentation results incorporated depth information are also obtained. The typical image segmentation results at the largest scale are shown in Fig 7. From the aspect of scene structure interpretation, the segmentation results get better and better after 1-3 iterations.



**Fig. 5.** Results for the predicted depthmap. The column a is original images, the column b is the results of Saxena's methods, the column c is the results of our methods, the column d is the groundtruth and the column e is the depth scale. The depths of sky regions in column c and d are denoted as zero. (Best viewed in color)



**Fig. 6.** Results for scene reconstruction. The 1<sup>st</sup> row is original images, the 2<sup>nd</sup> row is Saxena's scene reconstruction results, the 3<sup>rd</sup> row is Hoiem's results and the 4<sup>th</sup> row is our results.(Best viewed in color)



**Fig. 7.** Results for image segment. The 1<sup>st</sup> column is original image; the 2<sup>nd</sup> column is the initial segmentation under the largest scale; the 3<sup>rd</sup> -5<sup>th</sup> columns show the obtained segmentation results after 1-3 iterations.(Best viewed in color)

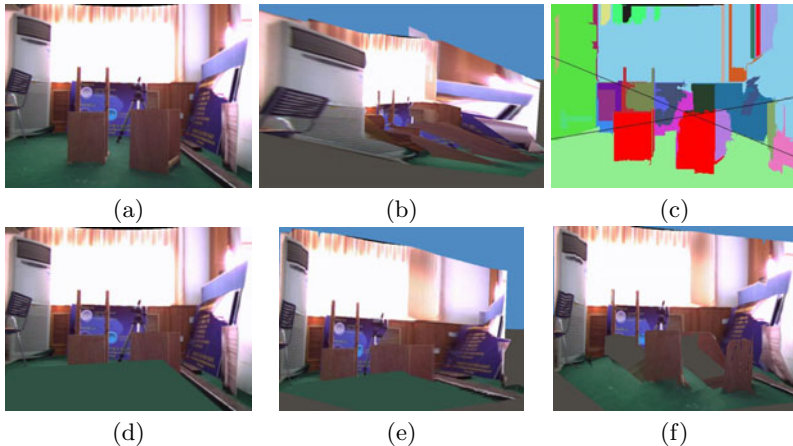


## 6 Scene Reconstruction with Fore-Object

As mentioned above, each segmentation region with the pixels having similar properties represents a coherent region in the scene. Thus, it will be sometimes failed when there are fore-objects in the scene. The example is shown as Fig. 8(a). A stool lies in front of the back wall with the similar color and texture. In the inferred 3D scene shown in Fig. 8(b), the stools are conjoint to the back wall, which is obviously wrong.

Under the on-the ground assumption, we propose a method to deal with the above problem. Actually, the fore-objects are most likely to be on the ground, rather than in it, especially at indoor environments. So we firstly find the ground region in an image. According to the initial scene reconstruction results, the edges of the ground region can easily be extracted and denoted as a set of lines  $l_1, l_2, \dots, l_n$ . Then the pixels surrounded by  $l_1, l_2, \dots, l_n$  are marked as ground region. As for the fore-object region in image, it is most likely to be intersected with the ground region, rather than to be included in it. So if a region has only a part of pixels marked as ground region, it can be regarded as fore-object region. The example of extracting the fore-object is shown as Fig. 8(c), black line is the edge of ground region and the red block is the fore-object.

Then the fore-object regions and the rest regions are dealt with respectively. As for the fore-object, it can be assumed to be vertical to ground since there is no more information about it. Based on the assumption, the depth is predicted according to projective geometry. As for the rest regions, the depth can be inferred by the methods described in section 4. Finally the scene reconstruction results are incorporated together. The experimental results are shown as Fig. 8(d,e,f).



**Fig. 8.** (a) Original image, (b) Results of scene reconstruction without the detecting of fore-object, (c) Results of detecting the edges of ground region (black lines) and extracting the fore-object (red block), (d, e) Results of scene reconstruction after extracting fore-object, (f) Final scene reconstruction results. (Best viewed in color)

## 7 Conclusion

Over the last few decades, great progresses have been made on the depth inferring and scene reconstruction from stereo, motion and other "triangulation" cues. However, the vast majority of this work has only used the geometric cues, but neglected the other depth cues contained in the image, such as texture, color, defocus and so on. In contrast, the recent research of monocular depth perception, such as Saxena et al's and Hoiem et al's work, is commendably supplementary to computer vision.

Inspired by these works, this paper presents a close-form iterative algorithm that utilizes the inter-correlated property between image segmentation and depth inferring. The algorithm can significantly improve segmentation and depth inferring by processing them alternately iteratively. Our algorithm firstly obtains the initial segmentation results by an EM-based algorithm. Then, a multi-scale Markov random field, trained by supervised learning, is used to model the relations between feature vectors and different depths. After the depth of each pixel are inferred, it is fed back to refine the segmentation results at the larger scales. This method can offer more global structure information without decreasing the number of the patches made up of 3d scene structure. The above processes are iteratively continued until the expected results are achieved. The experimental results show the validity of our algorithm. Moreover, the paper also extends the method to deal with the problem that infers depth of the scene with fore-objects. We believe that our algorithm can be used for many other applications in vision, such as robot navigation, building 3-d models of urban environments, and object recognition.

**Acknowledgments.** This research was funded by NSFC(No: 60705015, 60805019). We would like to thank Shuai Fang for her valuable contribution towards this research.

## References

1. Saxena, A., Chung, S.H., Ng, A.Y.: Learning depth from single monocular images. In: Neural Information Processing System (NIPS), vol. 18 (2005)
2. Saxena, A., Chung, S.H., Ng, A.Y.: 3-d depth reconstruction from a single still image. *International Journal of Computer Vision (IJCV)* 76, 53–69 (2007)
3. Saxena, A., Schulte, J., Ng, A.Y.: Depth estimation using monocular and stereo cues. In: International Joint Conference on Artificial Intelligence (IJCAI) (2007)
4. Saxena, A., Sun, M., Ng, A.Y.: Make3d: depth perception from a single still image. In: AAI Conference on Artificial Intelligence (AAAI) (2008)
5. Saxena, A., Sun, M., Ng, A.Y.: Make3d: Learning 3-d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 31, 824–840 (2008)
6. Hoiem, D., Efron, A.A., Hebert, M.: Automatic photo pop-up. In: International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH) (2005)
7. Hoiem, D., Efron, A.A., Hebert, M.: Geometric context from a single image. In: International Conference on Computer Vision (ICCV) (2005)

8. Hedau, V., Hoiem, D., Forsyth, D.: Recovering the spatial layout of cluttered rooms. In: International Conference on Computer Vision (ICCV) (2009)
9. Hoiem, D., Efros, A.A., Hebert, M.: Closing the loop on scene interpretation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2008)
10. Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. *International Journal of Computer Vision (IJCV)* 80, 3–15 (2008)
11. Malik, J., Rosenholtz, R.: Computing local surface orientation and shape from texture for curved surfaces. *International Journal of Computer Vision (IJCV)* 23, 149–168 (1997)
12. Malik, J., Perona, P.: Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America A* 7, 923–932 (1990)
13. Maki, A., Watanabe, M., Wiles, C.: Geotensity: Combining motion and lighting for 3d surface reconstruction. *International Journal of Computer Vision (IJCV)* 48, 75–90 (2002)
14. Zhang, R., Tsai, P.S., Cryer, J.E., Shah, M.: Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 21, 690–706 (1999)
15. Horry, Y., Anjyo, K.I., Arai, K.: Tour into the picture: using a spidery mesh interface to make animation from a single image. In: International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH) (1997)
16. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision (IJCV)* 47, 7–42 (2002)
17. Forsyth, D., Ponce, J.: *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference (2002)
18. Delage, E., Lee, H., Ng, A.: A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2006)
19. Sudderth, E.B., Torralba, A., Freeman, W.T., Willsky, A.S.: Depth from familiar objects: A hierarchical model for 3d scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2006)
20. Michels, J., Saxena, A., Ng, A.Y.: High speed obstacle avoidance using monocular vision and reinforcement learning. In: International Conference on Machine Learning (ICML) (2005)
21. Heitz, G., Gould, S., Saxena, A., Koller, D.: Cascaded classification models: Combining models for holistic scene understanding. In: Neural Information Processing Systems (NIPS) (2008)
22. Carson, C., Belongie, S., Greenspan, H., Malik, J.: Blobworld: image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 24, 1026–1038 (2002)
23. Garding, J., Lindeberg, T.: Direct computation of shape cues using scale-adapted spatial derivative operators. *International Journal of Computer Vision (IJCV)* 17, 163–191 (1996)
24. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. *International Journal of Computer Vision (IJCV)* 59, 167–181 (2004)

# Learning Shape Segmentation Using Constrained Spectral Clustering and Probabilistic Label Transfer

Avinash Sharma, Etienne von Lavante, and Radu Horaud

INRIA Grenoble Rhône-Alpes, Montbonnot Saint-Martin, France

**Abstract.** We propose a spectral learning approach to shape segmentation. The method is composed of a *constrained spectral clustering* algorithm that is used to supervise the segmentation of a shape from a training data set, followed by a *probabilistic label transfer* algorithm that is used to match two shapes and to transfer cluster labels from a training-shape to a test-shape. The novelty resides both in the use of the Laplacian embedding to propagate must-link and cannot-link constraints, and in the segmentation algorithm which is based on a learn, align, transfer, and classify paradigm. We compare the results obtained with our method with other constrained spectral clustering methods and we assess its performance based on ground-truth data.

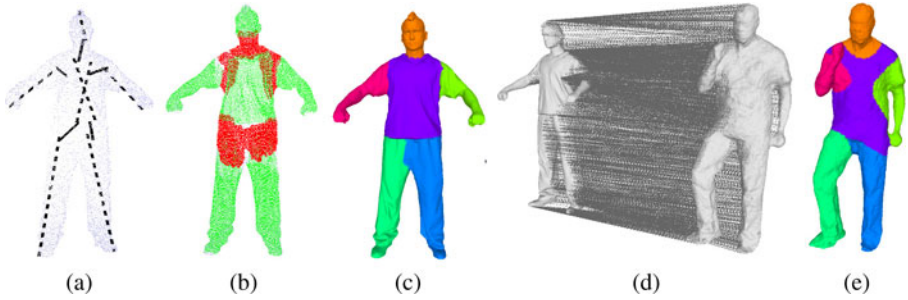
## 1 Introduction

In this paper we address the problem of segmenting shapes into their constituting parts with emphasis onto complex 3D articulated shapes. These shapes are difficult to describe in terms of their parts, e.g., body parts of humans, because there is a large variability within the same class of perceptually similar shapes. The reasons for this are numerous: changes in pose due to large kinematic motions, local deformations, topological changes, etc. Without loss of generality we will represent 3D shapes with meshes which can be viewed as both 2D discrete Riemannian manifolds and graphs. Therefore, shape segmentation can be cast into the problem of graph partitioning for which spectral clustering (SC) algorithms [1] provide tractable solutions.

Nevertheless, *unsupervised* spectral clustering algorithms will not always yield satisfactory shape segmentation results for the following reasons: Distances between vertices are only locally Euclidean (manifold structure), the graph has bounded connectivity (sparseness), and the number of edges meeting at each vertex is almost the same through the graph (regular connectivity). Manifoldness will exclude methods that need a fully-connected affinity matrix. While sparseness makes *shape-graphs* good candidates for Laplacian embedding [2,3], the usual spectral clustering assumptions do not hold in the case of regular connectivity. First, the Laplacian matrix of a shape-graph cannot be viewed as a slightly perturbed version of the ideal case<sup>1</sup>, namely a number of strongly connected components that are only weakly interconnected [1]. Second, there is no eigengap and hence there is no simple way to determine the number of clusters. Third, the eigenvectors associated with the smallest non-null eigenvalues cannot be viewed as relaxed indicator vectors [1].

---

<sup>1</sup> In the ideal case the between-cluster similarity cost is exactly 0.



**Fig. 1.** First stage: *Constrained spectral clustering* (CSC) which takes as input a mesh (or more generally a graph) together with a sparse set of must-link (dashed lines) and cannot-link (full lines) constraints (a). These constraints are propagated using the commute-time distance (b). Spectral clustering is applied to a modified graph Laplacian (c). Second stage: *Probabilistic label transfer* (PLT). Shape segmentation is performed via vertex-to-vertex matching (d) and label transfer (e).

In this paper we propose a learning approach to shape segmentation via a two-stage method, e.g., fig. 1. First we introduce a new constrained spectral clustering (CSC) algorithm which takes as input a shape-graph  $\mathcal{G}_{tr}$  from a *training* set.  $\mathcal{G}_{tr}$  contains unlabeled vertices as well as *must-link* and *cannot-link* constraints between pairs of vertices, fig. 1(a). These constraints are propagated, using the unnormalized Laplacian embedding and the commute-time distance (CTD), such that edge-weights corresponding to within-cluster connectivities are strengthened while those corresponding to between-cluster connectivities are weakened, fig. 1(b). This modified embedding yields improved shape segmentation results than the initial one, e.g., fig. 1(c), because it better fits into the theoretical requirements of spectral clustering [1].

Second, we consider shape alignment based on vertex-to-vertex graph matching as a way to probabilistically *transfer* labels from a training-set of segmented shapes to a test-set of unsegmented ones. We consider a shape-graph  $\mathcal{G}_{test}$  from a *test* set. The segmentation of  $\mathcal{G}_{test}$  is carried out via a new probabilistic label transfer (PLT) method that computes a point-to-point mapping between the embedding of  $\mathcal{G}_{tr}$  and  $\mathcal{G}_{test}$ , e.g., fig. 1(d). This completely unsupervised matching is based on [4,5] and allows to transfer labels from a segmented shape to an unsegmented one. Consequently, the vertices of  $\mathcal{G}_{test}$  can be classified using the segmentation trained with  $\mathcal{G}_{tr}$ , fig. 1(e). While the spectral graph matching is appealing [6], it adds an extra difficulty because of the ambiguity in the definition of spectral embeddings up to switching between eigenvectors corresponding to eigenvalues with multiplicity and changes in their sign [4,7]. This is particularly critical in the presence of symmetric shapes [8].

Unsupervised segmentation of articulated shapes is a well investigated problem and one can find a quantitative comparison of recent non-spectral methods in [9]. However, the spectral methods are natural choice for pose-invariant segmentation as they exploit the inherent manifold structure of the mesh representation to embed the shape in an isometric space. For the reasons already mentioned in the introduction, the results of simple spectral clustering (SC) are unsatisfactory ([1] for both a tutorial and

comprehensive study). Therefore, more recent methods, such as those by Reuter [10] and Zhang [11], also take the topological features of a shape in its embedded space into account and can achieve this way impressive segmentation results. However, these methods do not provide intuitive means to include constraints in a semi-supervised framework.

Regarding semi-supervised spectral methods, we distinguish between semi-supervised and constrained spectral clustering methods: With semi-supervised spectral methods we consider algorithms which attempt to find good partitions of the data given partial labels. In [12] labeled data information is propagated to nearby unlabeled data using a probabilistic label diffusion process, which needs an extra time parameter that must be specified in advance [13][14][7]. In [15] the labeled data are used to learn a classifier that is then used to sort the unlabeled data. These methods work reasonably well if there are sufficient labeled data or if the data can be naturally split into clusters. Furthermore, these methods were only applied to synthetic “toy” data and their extension to graphs that represent shapes may not be straightforward.

Constrained clustering methods use prior information under the form of pairwise must-link and cannot-link constraints, and were first introduced in conjunction with constrained K-means [16]. Subsequently, a number of solutions were proposed that consist in learning a distance metric that takes into account the pairwise relationships; This generally leads to convex optimization [17][18]. Since K-means is a ubiquitous post-processing step with almost any SC technique, it is tempting to replace it with constrained K-means. However, this does not take full advantage of the graph structure of the data where edges naturally encode pairwise relationships. Recently, metric learning has been extended to constrained spectral clustering leading to quadratic programming [19]. The semi-supervised kernel K-means method [20] incorporates constraints by adding reward and penalty terms to the cost function to be minimized.

Another way to incorporate constraints into spectral methods is to modify the affinity matrix of a graph using a simple rule: Edges between must-link vertex-pairs are set to 1 and edges between cannot-link pairs are set to 0 [21]. Despite its simplicity, this method is not easily extendible to our case due to graph sparsity: one has to add new edges (with value 1) and to remove some other edges. This will modify the graph’s topology and hence it will be difficult to use the segmentation learned on one shape in order to segment another shape.

All methods described above need a large number of constraints to work well, which is a major drawback, as it is desirable to work with a small set of sparse constraints. We note that the issue of constraint propagation is not well studied: The transitivity property of the must-link relationship has already been explored [22] but this cannot be used with the cannot-link relationship which is not transitive.

## 1.1 Paper Contributions

This paper has two contributions: A new constrained spectral clustering method that uses the unnormalized Laplacian embedding to propagate pairwise constraints and a modified Laplacian embedding to cluster the data, and a new shape segmentation method based on spectral graph matching and on a novel probabilistic label-transfer process.

We exploit the properties of the *unnormalized graph Laplacian* [31] which embeds the graph into an isometric space armed with a metric, namely the Euclidean *commute-time distance* (CTD) [23,14]. Unlike the diffusion maps that are parameterized by a discrete time parameter, which acts as a scale, [13], the CTD reflects the connectivity of two graph vertices: All possible paths of all lengths. We build on the idea of modifying the weighted adjacency matrix of a graph using instance level constraints on vertex-pairs [21]. We provide an explicit *constraint propagation* method that uses the Euclidean CTD to densify must-link and cannot-link relationships within small volumes lying between constrained data pairs. We show that the modified weighted adjacency matrix thus obtained can be used to construct a *modified Laplacian*. The latter respects the topology of the initial graph but with a distinct geometric structure that have the presence of dense *graph lumps*, which is a direct consequence of the constraint propagation process: This makes it particularly well suited for clustering.

We introduce a shape segmentation method based on a learn, align, transfer, and classify paradigm. This introduces an important innovation, namely that one can perform the training on one data-set and then classify a completely different data-set on the premise that the two sets are approximately isomorphic. Our probabilistic label transfer algorithm is robust to topological noise as we consider dense soft correspondences between two shapes.

We compare our CSC algorithm with several other methods recently proposed in the literature, and we evaluate it against ground-truth segmentations of both simulated and real shapes. We note that the existing CSC methods have not been applied to articulated shapes which are rather complex discrete Riemannian manifolds. Real shapes gathered with scanners and cameras are very challenging dataset. As already mentioned, these manifold data are very difficult to cluster due to the regularity of the associated graph.

## 2 Laplacian Embeddings and Their Properties

We consider an *undirected weighted graph*  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{A}\}$  where  $\mathcal{V}(\mathcal{G}) = \{v_1, \dots, v_n\}$  is the vertex set,  $\mathcal{E}(\mathcal{G}) = \{e_{ij}\}$  is the edge set, and the entries of the weighted adjacency matrix  $\mathbf{A}$  are:  $a_{ii} = 0$ ,  $a_{ij} > 0$  whenever two vertices are adjacent, i.e.,  $v_i \sim v_j$ , and  $a_{ij} = 0$  otherwise. In the case of 2D manifolds, a vertex  $v_i$  corresponds to a 3D point  $v_i$ . Let  $0 < a_{\min} \leq a_{ij} \leq a_{\max} \leq 1$ . Since our graphs correspond to a uniform surface discretization, it is realistic to assume that the weights vary within a small interval  $[a_{\min}, a_{\max}]$ . Without loss of generality we consider Gaussian weights i.e.  $a_{ij} = \exp(-d_{ij}^2/\sigma^2)$ .

We briefly recall the following definitions: the *degree matrix*  $\mathbf{D} = \text{Diag}[d_1 \dots d_n]$ , the *n-dimensional degree vector*  $\mathbf{d} = (d_1 \dots d_n)^\top$ , with  $d_i = \sum_{i \sim j} a_{ij}$ . The following Laplacian matrices are used in spectral clustering [11]: The *unnormalized Laplacian*  $\mathbf{L} = \mathbf{D} - \mathbf{A}$ , the *normalized Laplacian*  $\mathbf{L}_N = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2}$ , and the *random-walk Laplacian*  $\mathbf{L}_R = \mathbf{D}^{-1} \mathbf{L}$ . Both  $\mathbf{L}$  and  $\mathbf{L}_N$  are symmetric semi-positive definite, hence their eigenvalues are non-negative and their eigenvectors form an orthonormal vector basis of  $\mathbb{R}^n$ . From the similarity  $\mathbf{L}_R = \mathbf{D}^{-1/2} \mathbf{L}_N \mathbf{D}^{1/2}$  one can easily characterize the eigenspace of the random-walk graph Laplacian. It has been recently shown that both  $\mathbf{L}$  and  $\mathbf{L}_R$  are well suited for spectral clustering [11]. In this section we describe some

interesting properties of the unnormalized Laplacian which justify its use for both the tasks of clustering and of matching.

**The L-embedding.** Let  $\mathbf{L}\mathbf{u} = \lambda\mathbf{u}$ , denote  $\mathbf{\Lambda} = \text{Diag} [\lambda_2 \dots \lambda_{p+1}]$ , and let  $\mathbf{U} = [\mathbf{u}_2 \dots \mathbf{u}_{p+1}]$  be the  $n \times p$  matrix formed with the  $p$  smallest non-null eigenvectors of  $\mathbf{L}$ , hence  $\mathbf{U}^\top \mathbf{U} = \mathbf{I}_p$ . We have as well  $\lambda_1 = 0$  and  $\mathbf{u}_1 = \mathbf{1}$  (a vector with all entries equal to 1). The columns of  $\mathbf{U}$  form an orthonormal basis that span an embedded space  $\mathbb{R}^p \subset \mathbb{R}^n$  perpendicular to  $\mathbf{1}$ . Hence, we have the following property:

$$\sum_{j=1}^n \mathbf{u}_i(v_j) = 0, \forall i, 2 \leq i \leq p + 1 \tag{1}$$

where we introduced the notation  $\mathbf{u}_i(v_j)$  for the  $j$ -th entry of vector  $\mathbf{u}_i$  in order to emphasize that each eigenvector is an eigenfunction mapping the graph’s vertices onto real numbers. The Euclidean embedding of the graph’s nodes that we will use are the column vectors of the  $p \times n$  matrix  $\mathbf{X}$  defined by:

$$\mathbf{X} = \mathbf{\Lambda}^{-1/2} \mathbf{U}^\top = [\mathbf{x}_1 \dots \mathbf{x}_j \dots \mathbf{x}_n] \tag{2}$$

This is also known as the *commute-time embedding* [14]. From the orthonormality of the eigenvectors and from (I) we obtain:

$$-\lambda_i^{-1/2} < \mathbf{u}_i(v_j) < \lambda_i^{-1/2}, \forall j, 1 \leq j \leq n \tag{3}$$

**The  $\tilde{\mathbf{L}}$ -embedding.** So far we described the properties of spectral embeddings that correspond to graphs that contain only unlabeled vertices. As it will be explained in the next section, the presence of pairwise constraints could lead to a modified Laplacian embedding and in this paragraph we describe the rationale of this modified spectral representation. We suppose that pairwise constraints are provided and we consider one such vertex-pair. Two situations can occur: (i) the two vertices are adjacent or, more generally, (ii) the two vertices are connected by one or several graph paths. While the former situation leads to simply modifying the edge weights of the corresponding pairs, the latter is more problematic to implement because it involves some form of constraint propagation and it constitutes the topic of section 3. To summarize, the presence of constraints leads to modifying some of the edge weights in the graph. We denote the *modified adjacency matrix* with  $\tilde{\mathbf{A}}$ . We also obtain a modified degree matrix  $\tilde{\mathbf{D}}$  and a modified unnormalized Laplacian  $\tilde{\mathbf{L}}$ :

$$\tilde{\mathbf{L}} = \tilde{\mathbf{D}} - \tilde{\mathbf{A}} \tag{4}$$

This leads to modified Euclidean coordinates:

$$\tilde{\mathbf{X}} = \tilde{\mathbf{\Lambda}}^{-1/2} \tilde{\mathbf{U}}^\top = [\tilde{\mathbf{x}}_1 \dots \tilde{\mathbf{x}}_j \dots \tilde{\mathbf{x}}_n] \tag{5}$$

The initial graph can therefore be represented with two different embeddings, the exact geometry of the embedded space depending on the edge weights. Notice, that there is a one-to-one correspondence between the columns of  $\mathbf{X}$  and of  $\tilde{\mathbf{X}}$ .

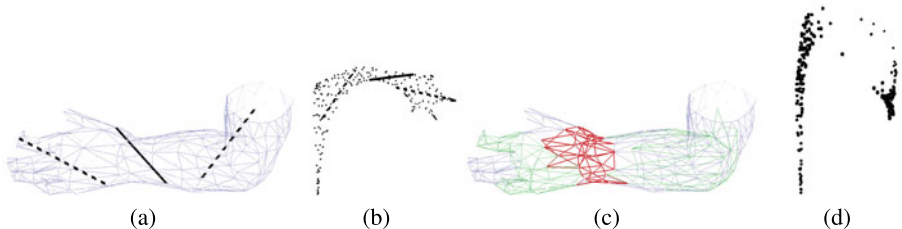


### 3 Propagating Pairwise Constraints

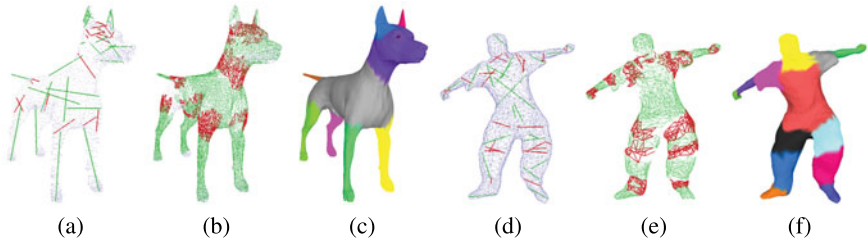
In a constrained clustering task instance-level constraints are available. In practice, it is convenient to be able to cope with a sparse set of constraints. The counterpart is that they are not easily exploitable: propagating these constraints over a manifold (or more generally over a graph) is problematic. In this section we describe a constraint propagation method that uses the  $L$ -embedding and the associated Euclidean *commute-time distance* (CTD). As already mentioned, must-link and cannot-link constraints were successfully incorporated in several variant of the K-means algorithm [16][17][18]. However, these methods did not incorporate constraint propagation. Rather than modifying the K-means step of spectral clustering, we incorporate a constraint-propagation process directly into the  $L$ -embedding, thus fully exploiting the properties outlined in the previous section.

Consider a subset of the set of graph vertices  $\mathcal{S} = \{\bar{v}_i\}, \mathcal{S} \subset \mathcal{V}$  from which we build two sets of constraints: A must-set  $\mathcal{M} \subset \mathcal{S} \times \mathcal{S}$  and a cannot-set  $\mathcal{C} \subset \mathcal{S} \times \mathcal{S}$ . Vertex pairs from the must-set should be assigned to the same cluster while vertex pairs from the cannot-set should be assigned to different clusters. Notice that the cardinality of these sets is independent of the final number of clusters. Also, it is necessary neither to provide must links for all the clusters, nor to provide cannot links across all cluster pairs. A straightforward strategy for enforcing these constraints consists in modifying the weights  $a_{ij}$  associated with *adjacent* vertex-pairs that belong either to  $\mathcal{M}$  or to  $\mathcal{C}$ , such that  $a_{ij}$  is replaced with  $\tilde{a}_{ij} = 1$  if  $(\bar{v}_i, \bar{v}_j) \in \mathcal{M}$  and  $\tilde{a}_{ij} = \varepsilon$  if  $(\bar{v}_i, \bar{v}_j) \in \mathcal{C}$ , where  $\varepsilon$  is a small positive number. We recall that  $0 < a_{\min} \leq a_{ij} \leq a_{\max} \leq 1$ . Notice that for graphs corresponding to regular meshes, the edge-weight variability is small.

Since the set  $\mathcal{S}$  is composed of sparsely distributed vertices, the pairs  $(\bar{v}_i, \bar{v}_j)$  do not necessarily correspond to adjacent vertices. Hence, one has to propagate the initial must-link and cannot-link constraints to nearby vertex pairs. We propose to use the commute-time distance (CTD) already mentioned. The CTD is a well known quantity in Markov chains [24]. For undirected graphs, it corresponds to the average number of (weighted) edges that it takes, starting at vertex  $v_i$ , to randomly reach vertex  $v_j$  for the first time and go back. The CTD has the interesting property that it decreases as the number of paths connecting the two nodes increases and when the lengths of the paths



**Fig. 2.** Propagating constraints. (a): Constraint placement onto the initial graph, two must-links (dashed lines) and one cannot-link; (b): The  $L$ -embedding used for constraint propagation. (c): The propagated constraints are shown on the graph. (d): The new embedding obtained with the modified Laplacian  $\tilde{L}$ .



**Fig. 3.** The CSC algorithm applied to the the *dog* and to the *flashkick* data (Note: unlike the results in Table I, we seek here for *flashkick* 14 segments). Initial graphs and manually placed constraints (a), (d); Constraint propagation (b), (e); Final clustering results (c), (f).

decrease. We prefer the CTD to the shortest-path geodesic distance in the graph because it captures the connectivity structure of a small graph volume rather than a single path between two vertices. The CTD is the integral of the diffusion distances over all times. Hence, unlike the latter, the former does not need the free parameter  $t$  to be specified [13][14][7]. Indeed, the scale parameter introduces an additional difficulty because different vertex-pairs may need to be processed at different scales. The commute-time distance [23] between two vertices is an Euclidean metric and it can be written in closed form using the  $\mathbf{L}$ -embedding, i.e., eq. (2):

$$d_{\text{CTD}}^2(v_i, v_j) = \|\mathbf{x}_i - \mathbf{x}_j\|^2 \quad (6)$$

The CTD will allow us to propagate must-link and cannot-link constraints within small graph volumes, e.g., fig. 2

We briefly describe the **propagation of must-link constraints**. For each pair  $(\tilde{v}_i, \tilde{v}_j) \in \mathcal{M}$  with embedded coordinates  $\tilde{\mathbf{x}}_i$  and  $\tilde{\mathbf{x}}_j$ : We consider the hypersphere centered at  $(\tilde{\mathbf{x}}_i + \tilde{\mathbf{x}}_j)/2$  with diameter given by (6) and we build a subset  $\mathbf{X}_s \subset \mathbf{X}$  that contains embedded vertices lying in this hypersphere. We build a subgraph  $\mathcal{G}_s \subset \mathcal{G}$  having as vertices the set  $\mathcal{X}_s = \{v_i\}_{i=1}^r$  corresponding to  $\mathbf{X}_s$ . Finally, we modify the weights  $a_{ij}$  of the edges of  $\mathcal{G}_s$ :  $\tilde{a}_{ij} = 1$ . There is an equivalent procedure for the **propagation of cannot-link constraints**. In order to preserve the topology of the modified graph, in this case the weights are set to a small positive number, i.e., the modified weight of a cannot-edge is  $\tilde{a}_{ij} = \varepsilon$ . Hence the proposed CSC algorithm, fig. 3

---

### Algorithm 1. Constrained Spectral Clustering (CSC)

---

**input** : Unnormalized Laplacian  $\mathbf{L}$  of a shape-graph  $\mathcal{G}$ , a must-link set  $\mathcal{M}$ , a cannot-link set  $\mathcal{C}$ , the number of cluster  $k$  to construct.

**output** : A set of binary variables  $\Delta = \{\delta_{il}\}$  assigning a cluster label  $l$  to each graph vertex  $v_i$ .

- 1: Compute the  $\mathbf{L}$ -embedding of the graph using the  $p$  first non-null eigenvalues and eigenvectors of  $\mathbf{L}$ ,  $p \geq k$ .
  - 2: Propagate the  $\mathcal{M}$  and  $\mathcal{C}$  constraints, modify the adjacency matrix of  $\mathcal{G}$  and build the modified Laplacian  $\tilde{\mathbf{L}}$  using eq. (4).
  - 3: Compute the  $\tilde{\mathbf{L}}$ -embedding using the  $k$  first non-null eigenvalues and eigenvectors of  $\tilde{\mathbf{L}}$ .
  - 4: Assign a cluster label  $l$  to each graph vertex  $v_i$  by applying K-means to the points  $\tilde{\mathbf{X}}$  in eq. (5).
-

## 4 Shape Segmentation via Label Transfer

The CSC algorithm that we just described is applied to a shape-graph  $\mathcal{G}_{\text{tr}}$  such that the latter is segmented into  $k$  clusters. Given a second shape  $\mathcal{G}_{\text{test}}$  we wish to use the segmentation result obtained with  $\mathcal{G}_{\text{tr}}$  to segment  $\mathcal{G}_{\text{test}}$ . Therefore, the segmentation of  $\mathcal{G}_{\text{test}}$  can be viewed as an inference problem, where we seek a cluster label for each one of its vertices conditioned by the segmentation of  $\mathcal{G}_{\text{tr}}$ .

We formulate this label inference problem in a probabilistic framework and adopt a generative approach where we model the conditional probability of assigning a label to a test shape vertex. More formally, let  $\mathbf{X}^{\text{tr}}$  and  $\mathbf{X}^{\text{test}}$  be the L-embeddings of the two shapes with  $n$  and  $m$  vertices respectively, i.e., eq. (2). We introduce three sets of hidden variables:  $S = \{s_1, \dots, s_m\}$  which assign each *test-shape* vertex to its cluster,  $R = \{r_1, \dots, r_n\}$  which assign each *train-shape* vertex to its cluster, and  $Z = \{z_1, \dots, z_m\}$ , which assign a test-shape vertex to a train-shape vertex. Then the posterior probability of assigning a cluster label  $l \in \{1, \dots, k\}$  to a test-shape vertex  $\mathbf{x}_i^{\text{test}} \in \mathbf{X}^{\text{test}}$  can be written as:

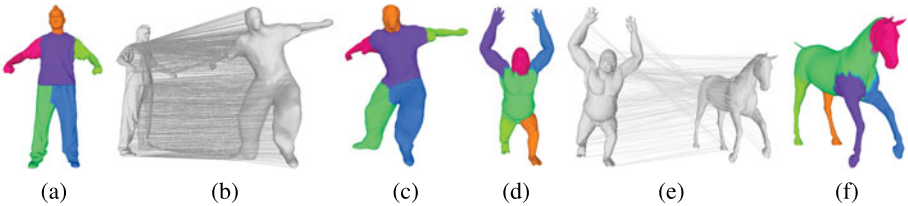
$$P(s_i = l | \mathbf{x}_i^{\text{test}}) = \sum_{j=1}^n P(r_j = l | \mathbf{x}_j^{\text{tr}}) P(z_i = j | \mathbf{x}_i^{\text{test}}), \quad (7)$$

Here,  $P(r_j = l | \mathbf{x}_j^{\text{tr}})$  is the posterior probability of assigning a label  $l$  to a train-shape vertex  $\mathbf{x}_j^{\text{tr}}$ , conditioned by the train-shape vertex. Similarly,  $P(z_i = j | \mathbf{x}_i^{\text{test}})$  is the posterior probability of assigning train-shape vertex  $\mathbf{x}_j^{\text{tr}}$  to test-shape vertex  $\mathbf{x}_i^{\text{test}}$  and can be termed as *soft assignment*. We propose to replace the posteriors  $P(r_j = l | \mathbf{x}_j^{\text{tr}})$  with hard assignments, namely the output of the CSC algorithm:

$$P(r_j = l | \mathbf{x}_j^{\text{tr}}) = \delta_{jl} \quad (8)$$

The estimation of the posteriors  $P(z_i = j | \mathbf{x}_i^{\text{test}})$  is an instance of graph matching in the spectral domain which is a difficult problem in its own right, especially in the presence of switches between eigenvectors and changes in their sign. The graph/shape matching task is further complicated when the two graphs are not isomorphic and when they have different numbers of vertices.

We adopted the articulated shape matching method proposed in [45] to obtain these *soft assignments*. This method proceeds in two steps. The first step uses the histograms of the  $k$  first non-null eigenvectors of the *normalized* Laplacian matrix to find an alignment between the Euclidean embeddings of two shapes. The second step registers the



**Fig. 4.** Clustering obtained with CSC (a), (d); vertex-to-vertex probabilistic assignment between two shapes (b), (e); The result of segmenting the second shape based on label transfer (c), (f)

two embeddings using the expectation-maximization (EM) algorithm and selects the best vertex-to-vertex assignment based on the maximum a posteriori probability (MAP) of a vertex from one shape to be assigned to a vertex from the other shape. In order to fit to our methodological framework, we introduce two important modifications to the technique described in [4]:

1. We use the unnormalized Laplacian. This is justified by the properties of the  $L$ -embeddings which were described in detail in section 2. In particular, the property (3) facilitates the task of comparing the histograms of two eigenvectors.
2. We do not attempt to find the best one-to-one assignments based on the MAP criterion. Instead, we keep all the assignments and hence we rely on *soft* rather than *hard* assignments.

The resulting shape matching algorithm will output the desired posterior probabilities  $P(z_i = j | \mathbf{x}_i^{\text{test}}) = p_{ij}, \forall 1 \leq i \leq m$ . From (7) and (8) we obtain the following expression that probabilistically assigns a vertex of  $\mathcal{G}_{\text{test}}$  to a cluster of  $\mathcal{G}_{\text{tr}}$ :

$$\gamma_{il} = \arg \max_{1 \leq l \leq k} \sum_{j=1}^n p_{ij} \delta_{jl} \tag{9}$$

This corresponds to the maximum posterior probability of a test-shape vertex to be assigned to a train-shape cluster conditioned by the test-shape vertex and by the train-shape-to-test-shape soft assignments of vertices. The proposed segmentation method is summarized in algorithm 2. Fig. 4 illustrates the PLT method on two examples.

---

**Algorithm 2. Probabilistic Label Transfer (PLT)**

---

- input** :  $L$ -embeddings  $\mathbf{X}^{\text{tr}}$  and  $\mathbf{X}^{\text{test}}$  of train and test shape-graphs  $\mathcal{G}_{\text{tr}}$  and  $\mathcal{G}_{\text{test}}$ ; a set of binary variables  $\Delta = \{\delta_{jl}\}$  assigning a cluster label  $l$  to each vertex  $\mathbf{x}_j^{\text{tr}} \in \mathbf{X}^{\text{tr}}$ .
- output** : A set of binary variables  $\Gamma = \{\gamma_{il}\}$  assigning a cluster label  $l$  to each vertex  $\mathbf{x}_i^{\text{test}} \in \mathbf{X}^{\text{test}}$ .
- 1: Align two  $L$ -embeddings  $\mathbf{X}^{\text{tr}}$  and  $\mathbf{X}^{\text{test}}$  using the histogram alignment method [4].
  - 2: Compute the posterior probability  $p_{ij}$  of assigning each test graph vertex  $\mathbf{x}_i^{\text{test}}$  to every train graph vertex  $\mathbf{x}_j^{\text{tr}}$  using the EM based rigid point registration method proposed in [5].
  - 3: Find the cluster label  $l$  for each test graph vertex  $\mathbf{x}_i^{\text{test}}$  using the eq.(9).
- 

## 5 Experiments and Results

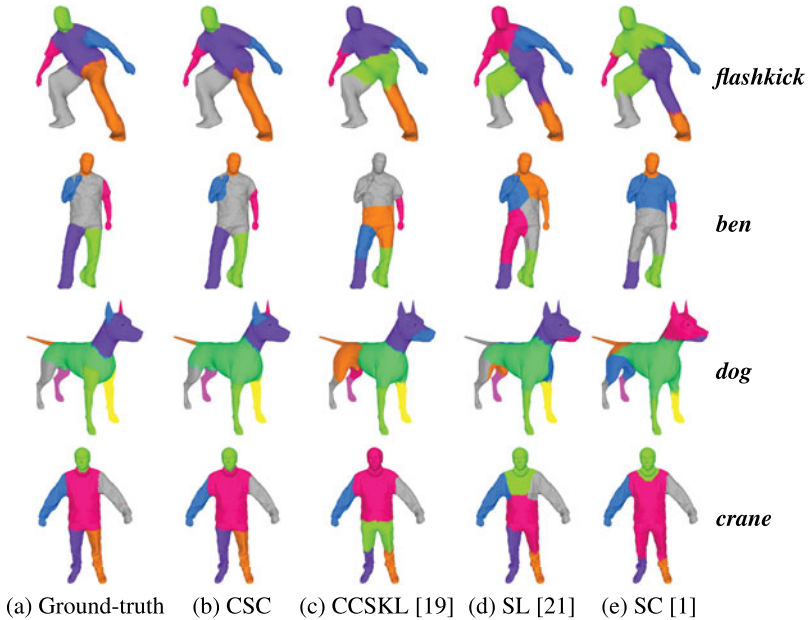
We evaluated the performance of our approach on 3D meshes, consisting of both synthetic [2] and real articulated shapes [3] having a wide range of variability in terms of mesh topology, kinematic poses, noise and scale. Particularly, the data acquired by multi-camera systems are non-uniformly sampled and there are major topological changes in

<sup>2</sup> [http://tosca.cs.technion.ac.il/book/resources\\_data.html](http://tosca.cs.technion.ac.il/book/resources_data.html)  
<sup>3</sup> [http://people.csail.mit.edu/draniel/mesh\\_animation/index.html](http://people.csail.mit.edu/draniel/mesh_animation/index.html)  
<http://4drepository.inrialpes.fr/public/datasets>  
<http://www.ee.surrey.ac.uk/CVSSP/VisualMedia/VisualContentProduction/Projects/SurfCap>

between the various kinematic poses, e.g., fig. 1(c). We have generated manual segmentations of all the employed meshes as a ground truth for the quantitative evaluation of our approach. As a consequence of this, one-to-one correspondences between ground-truth and our results are available. Therefore, the standard statistical error measures like the true positives  $e_i^{tp}$ , the false negatives  $e_i^{fn}$  and the false positives  $e_i^{fp}$  can be easily computed for each segmentation and for each cluster  $i$ . From these measures we derive the true positive rate  $m_i^{tpr}$  (recall) and positive predictive value  $m_i^{ppv}$  (precision) for every cluster:  $m_i^{tpr}$  gives for each cluster  $i$  the percentage of vertices which have been correctly identified from the ground truth, and  $m_i^{ppv}$  gives for each identified cluster the percentage of vertices which actually truly belong to this cluster. Using these two measures, we tabulate the overall performance of our segmentation results by computing the mean over all clusters of each shape mesh. We can define recall and precision as:

$$\bar{m}^{tpr} = \sum_{i=1}^k \frac{e_i^{tp}}{e_i^{tp} + e_i^{fn}}, \quad \bar{m}^{ppv} = \sum_{i=1}^k \frac{e_i^{tp}}{e_i^{tp} + e_i^{fp}}$$

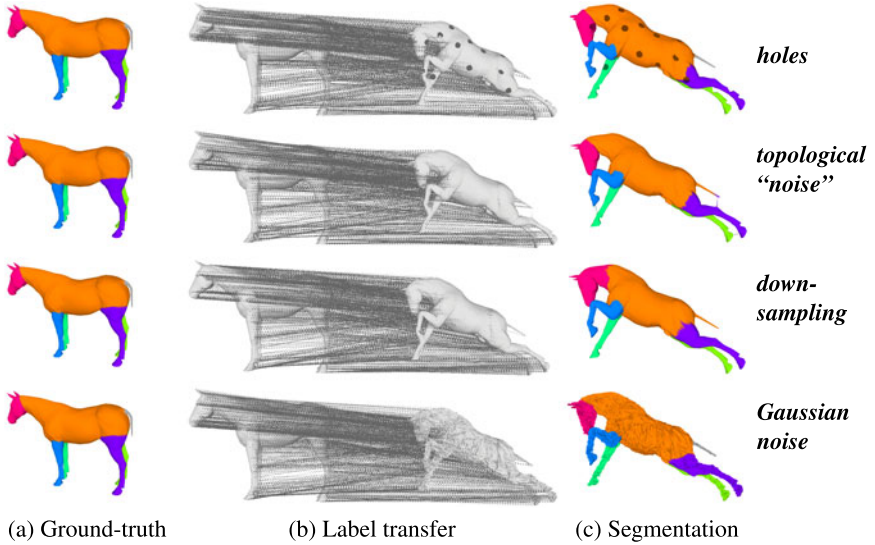
with  $k$  being the total number of clusters on the evaluated mesh. To maintain the independence of the ground truth from the test data, the manual segmentation and constraint placement for the tested algorithms were performed by different persons. We performed two sets of experiments. First, we evaluate the segmentation performance of the CSC algorithm described in section 3 against two other constrained spectral



**Fig. 5.** Manual segmentation (ground-truth), results obtained with our algorithm (CSC) and results obtained with three other methods

**Table 1.** Comparison of constrained spectral clustering algorithms

	$ \mathcal{V} $	$k$	$ \mathcal{M} $	$ \mathcal{C} $	CSC		CCSKL [19]		SL [21]		SC [1]	
					$\bar{m}^{tpr}$	$\bar{m}^{ppv}$	$\bar{m}^{tpr}$	$\bar{m}^{ppv}$	$\bar{m}^{tpr}$	$\bar{m}^{ppv}$	$\bar{m}^{tpr}$	$\bar{m}^{ppv}$
<i>dog</i>	3400	9	28	19	0.8876	0.9243	0.5215	0.6239	0.4342	0.5644	0.5879	0.6825
<i>crane</i>	10002	6	9	8	0.9520	0.9761	0.6401	0.7952	0.8673	0.7905	0.7818	0.8526
<i>handstand</i>	10002	6	7	5	0.9659	0.9586	0.6246	0.7691	0.6475	0.7246	0.7584	0.9248
<i>flashkick 89</i>	1501	6	18	5	0.9279	0.9629	0.5898	0.7539	0.5412	0.5984	0.6207	0.7376
<i>ben</i>	16982	6	7	5	0.9054	0.9563	0.4002	0.5888	0.6434	0.6084	0.5587	0.6494

**Fig. 6.** Segmentation results with synthetic meshes which have been corrupted in various ways

clustering algorithms; Second, we evaluate the probabilistic label-transfer method described in section 4.

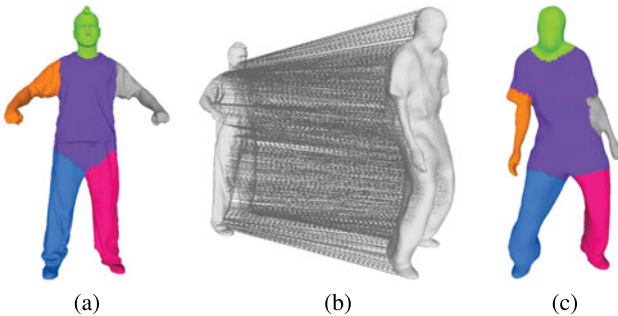
We compared our CSC algorithm with the *constrained clustering by spectral kernel learning* (CCSKL) method [19], and with the *spectral learning* (SL) method [21]. For completeness we also provide a comparison with the spectral clustering algorithm (SC) based on the random-walk graph Laplacian. Our implementations of these methods were duly checked with their respective cited results. With all these constrained spectral clustering methods the same set of constraints was used as well as the same number of clusters (the latter varies from one data set to another). The *normalized* SC algorithm that we implemented corresponds to the second algorithm in [1]: it applies K-means to the unnormalized Laplacian embedding, i.e., eq. (2) and it corresponds to steps 3 and 4 of our own CSC algorithm. A summary of these results can be found in Table 1 and fig. 5. The most surprising result is that, except for the “Crane” data and with SL, both CCSKL and SL could not significantly improve over the unsupervised SC algorithm, despite the side-information available to guide the segmentation. The CCSKL

**Table 2.** Summary of evaluating the PLT algorithm

Results for several meshes (I)						Results for corrupted horse meshes (II)				
$\mathcal{G}_{tr}$	$\mathcal{G}_{test}$	$ \mathcal{V}_{tr} $	$ \mathcal{V}_{test} $	$\bar{m}^{tpr}$	$\bar{m}^{ppv}$	transform	$ \mathcal{V}_{tr} $	$ \mathcal{V}_{test} $	$\bar{m}^{tpr}$	$\bar{m}^{ppv}$
<i>ben</i>	<i>handstand</i>	16982	10002	0.9207	0.9594	<i>topology</i>	19248	19248	0.9668	0.9642
<i>handstand</i>	<i>ben</i>	10002	16982	0.9672	0.9462	<i>sampling</i>	19248	8181	0.8086	0.9286
<i>flashkick 50</i>	<i>flashkick 89</i>	1501	1501	0.8991	0.9248	<i>noise</i>	19248	19248	1.0	1.0
<i>gorilla</i>	<i>horse</i>	2038	3400	0.8212	0.8525	<i>holes</i>	19248	21513	0.9644	0.9896

algorithm fails to improve over SC with our mesh data. Indeed, both assume that there are natural partitions (subgraph) in the data which are only weakly inter connected. Therefore, CCSKL only globally stretches each eigenvector in the embedded space to satisfy the constraints, without any local effect of these constraints on the segmentation. The SL algorithm can barely improve over the SC results as it requires a large number of constraints. With our method the placement of the *cannot-link* constraints is crucial. Although our method needs only a sparse set of constraints, the number of constraints increases (still number of constraints  $\ll |\mathcal{V}|$ ) if the desired segmentation is not consistent with the graph topology, e.g., fig. 3(d).

In the second experiment, we evaluate the performance of our probabilistic label transfer (PLT) method. In all these examples, we consider two different shapes, one from the training set and one from the test set. First we apply the CSC algorithm to the train-shape and then we apply the PLT algorithm to the test-shape. Fig. 1 shows an example of PLT between two different shapes and in the presence of significant topological changes: the right arm of Ben, (e), touches the torso. Fig. 4 show additional results which are quantified on Table 2(I). We also evaluate the robustness of PLT with respect to various mesh corruptive transformations, such as holes, topological noise, etc. Fig. 6 and Table 2(II) shows the segmentation results obtained by transferring labels from the original horse mesh to its corrupted instances. We obtain zero error if the corruptive transformation does not change the triangulation of the mesh as in the case of Gaussian noise. In fig. 7 we show the segmentation obtained with PLT where

**Fig. 7.** Clustering obtained with CSC (a); vertex-to-vertex probabilistic assignment between two shapes (b); The result of segmenting the second shape based on label transfer (c)

the test shape fig. 7(c) significantly differs from the training shape fig. 7(a) due to large acquisition noise (see the left hand merged with the torso).

## 6 Conclusions

We proposed a novel framework for learning shape segmentation. We made two contributions: (1) we proposed to use the unnormalized Laplacian embedding and the commute-time distance to diffuse sparse pairwise constraints over a graph and to design a new constrained spectral clustering algorithm, and (2) we proposed a probabilistic label transfer algorithm to segment an unknown test-shape by assigning labels between an already segmented train-shape and a test-shape. We perform extensive testing of both the CSC and the PLT algorithms on real and synthetic meshes. We compare our shape segmentation method with recent constrained/semi-supervised spectral clustering methods which were known to outperform unsupervised SC algorithms. However, we found it difficult to adapt these existing constrained clustering methods to the problem of shape segmentation. This is due to the fact that, unlike our method, they do not explicitly take into account the properties inherently associated with meshes, such as sparsity and regular connectivity.

## References

1. von Luxburg, U.: A tutorial on spectral clustering. *Statistics and Computing* 17, 395–416 (2007)
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* 15, 1373–1396 (2003)
3. Spielman, D.A., Teng, S.H.: Spectral partitioning works: Planar graphs and finite element meshes. *Linear Algebra and its Applications* 421, 284–305 (2007)
4. Mateus, D., Horaud, R., Knossow, D., Cuzzolin, F., Boyer, E.: Articulated shape matching using Laplacian eigenfunctions and unsupervised point registration. In: *CVPR* (2008)
5. Horaud, R., Forbes, F., Yguel, M., Dewaele, G., Zhang, J.: Rigid and articulated point registration with expectation conditional maximization. *IEEE PAMI* (2010) (in press)
6. Bronstein, A., et al.: Shrec 2010: Robust correspondence benchmark. In: *Eurographics Workshop on 3D Object Retrieval* (2010)
7. Bronstein, A., Bronstein, M., Kimmel, R., Mahmoudi, M., Sapiro, G.: A Gromov-Hausdorff framework with diffusion geometry for topologically robust non-rigid shape alignment. *IJCV* 89, 266–286 (2010)
8. Ovsjanikov, M., Sun, J., Guibas, L.: Global intrinsic symmetries of shapes. *Computer Graphics Forum* 27, 1341–1348 (2008)
9. Chen, X., Golovinskiy, A., Funkhouser, T.: A benchmark for 3d mesh segmentation. In: *ACM Transactions on Graphics (SIGGRAPH)* (2009)
10. Reuter, M.: Hierarchical shape segmentation and registration via topological features of laplace-beltrami eigenfunctions. *IJCV* 89, 287–308 (2010)
11. Liu, R., Zhang, H.: Mesh segmentation via spectral embedding and contour analysis. *Computer Graphics Forum* 26, 385–394 (2007)
12. Szummer, M., Jaakkola, T.: Partially labeled classification with Markov random walks. In: *NIPS* (2002)



13. Coifman, R., Lafon, S.: Diffusion maps. *Applied and Computational Harmonic Analysis* 21, 5–30 (2006)
14. Qiu, H., Hancock, E.R.: Clustering and embedding using commute times. *IEEE PAMI* 29, 1873–1890 (2007)
15. Belkin, M., Niyogi, P.: Semi-supervised learning on Riemannian manifold. *Machine Learning* 56 (2004)
16. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S.: Constrained k-means clustering with background knowledge. In: *ICML* (2001)
17. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.J.: Distance metric learning with application to clustering with side-information. In: *NIPS* (2002)
18. Bilenko, M., Basu, S., Mooney, R.J.: Integrating constraints and metric learning in semi-supervised clustering. In: *ICML* (2004)
19. Li, Z., Liu, J.: Constrained clustering by spectral kernel learning. In: *ICCV* (2009)
20. Kulis, B., Basu, S., Dhillon, I., Mooney, R.: Semi-supervised graph clustering: a kernel approach. *Machine Learning* 74, 1–22 (2009)
21. Kamvar, S.D., Klein, D., Manning, C.D.: Spectral learning. In: *IJCAI* (2003)
22. Yu, S.X., Shi, J.: Segmentation given partial grouping constraints. *IEEE PAMI* 26, 173–183 (2004)
23. Fouss, F., Pirotte, A., Renders, J., Saerens, M.: Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation. *IEEE KDE* 19, 355–369 (2007)
24. Grinstead, C.M., Snell, J.L.: *Introduction to Probability*. American Mathematical Society, Providence (1998)

# Weakly Supervised Shape Based Object Detection with Particle Filter

Xingwei Yang and Longin Jan Latecki

Dept. of Computer and Information Science  
Temple University, Philadelphia, 19122, USA  
{xingwei.yang,latecki}@temple.edu

**Abstract.** We describe an efficient approach to construct shape models composed of contour parts with partially-supervised learning. The proposed approach can easily transfer parts structure to different object classes as long as they have similar shape. The spatial layout between parts is described by a non-parametric density, which is more flexible and easier to learn than commonly used Gaussian or other parametric distributions. We express object detection as state estimation inference executed using a novel Particle Filters (PF) framework with static observations, which is quite different from previous PF methods. Although the underlying graph structure of our model is given by a fully connected graph, the proposed PF algorithm efficiently linearizes it by exploring the conditional dependencies of the nodes representing contour parts. Experimental results demonstrate that the proposed approach can not only yield very good detection results but also accurately locates contours of target objects in cluttered images.

## 1 Introduction

Object recognition, detection, and localization in real images is a major problem in Computer Vision since its beginning. In the last few years, the majority of existing methods use simple relations of local image patches as basic features, e.g., [24,3]. They can perform very well on high textured objects, but they are unable to identify parts of deformable objects nor precisely localize their boundaries in images. The main reason is that the model fails to represent all available information [25]. However, an improved, richer representation of deformable objects is only useful when it is accompanied by efficient techniques for performing inference and learning [26]. Thus, progress in this area requires to simultaneously develop more powerful representations together with efficient inference algorithms.

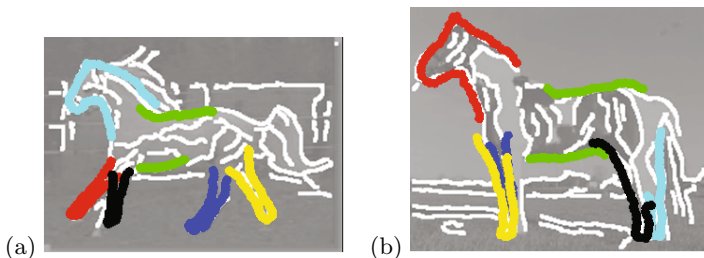
In this paper, we propose a single layer fully connected graph to model shape of deformable objects. Each node in the graph is a state variable, which consists of the position and the corresponding part. The relation between nodes is long range and not limited to direct spatial proximity. Our model can be interpreted as a generative prior for the configuration of the state variables. Since our graph is fully connected, we do not need to learn its structure, which simplifies

the learning significantly. We only need to learn representation of the nodes and their pairwise relations. Since the number of pairwise relations is large, and most of them are not used in our inference process, we do not learn the pairwise relations explicitly. Instead, we learn a representation that allows us to dynamically construct the pairwise relations needed in the inference process.

In our model graph, the nodes represent contour parts and their position in a given shape class. They are learned automatically with partially-supervised learning. While many state-of-the-art approaches construct part models manually [18,27], we limit manual labeling to a single contour. In our approach, only one silhouette is manually decomposed into visual parts in advance. Then, the part decomposition is automatically transferred to silhouettes not only in the same class but also in different classes with similar shape by shape matching. To deal with non-rigid objects, we use Inner Distance Shape Context (IDSC) introduced in [17]. The constructed part bundles (see §3) with proper position in the exemplar shapes form the nodes in the model graph. The relations between the nodes represent the spatial layout between parts. It is described by nonparametric density estimation, which has better discriminative power than methods based on unimodal distributions modeled as Gaussians, e.g., [5,23]. To make the learnt model graph representative, we use the well designed exemplar based clustering by Affinity Propagation [8] to select a set of candidate silhouettes as exemplars for our model learning approach.

According to [26], there are no known algorithms for performing inference for densely connected flat models, e.g., the performance of Belief Propagation (BP) is known to degrade for representations with many closed loops. To address this issue, we propose a Markov chain Monte Carlo (MCMC) approach that is able to efficiently infer the values of the state variables representing nodes of our fully connected model graph. The proposed MCMC approach is based on Particle Filter (PF), but it differs fundamentally, since unlike the standard PF framework, our PF framework can infer an order of random variable (RVs). The inferred order follows the most informative paths in the graph. Thus, we use PF to linearize the structure of the graph, which allows us to avoid the problem of loops. Each particle may explore a different node order in this linearization, which corresponds to the order of contour parts. This fact is illustrated by two different detection examples shown in Fig. 1, where the PF order of detected contour parts is color coded. This property makes our algorithm different from other PF based method [13,12]. As can be seen by examining the relative position of consecutive parts, the proposed inference is not limited to direct spatial proximity of the parts. This fact sets our approach apart from existing approaches, e.g., [26,14].

In order to show the advantages of the proposed approach, we test our method on three widely used data sets, Weizmann horses [2], the ETHZ [6], and the cow dataset from the PASCAL Object Recognition Database Collection (TU Darmstadt Database [16]). Our results measured by bounding box intersection are comparable to state-of-the-art methods. Also, we perform very well in the accuracy of boundary localization, which is evaluated by a recently proposed measure in [7].



**Fig. 1.** Examples of two different inferred orders of detected contour parts. Colors represent the order, which is 1=red, 2=cyan, 3=blue, 4=green, 5=yellow, and 6=black.

## 2 Related Work

Ferrari et al. [7] propose to learn the model from real images with weakly supervision. Given the bounding boxes, the model is considered as the common pattern of objects in the same class. With the same intuition, Lee and Grauman [15] also treat the common pattern in a class as the model. However, their method is totally unsupervised. To utilize the already learnt information, Stark et al. [23] transfer information of the learnt model to better study the new model by a probabilistic framework. They have very similar intuition with our method. However, ours are quite different from theirs. We transfer the structure information by pure shape matching without any statistics. Their method is mainly based on the probabilistic model they construct.

To detect objects in the cluttered image, Ferrari et al. [6] use kAS with Hough voting to estimate the position of objects. Ommer and Malik [20] propose a novel Hough voting strategy to overcome the problem of scales. Zhu et al. [27] treat the detection as a set-to-set matching problem between segments. They simplify the problem into linear programming to reduce the complexity. Ravishankar et al. [21] propose a multi-stage method with manually deformed model. Similar to ours, Trinh and Kimia also learn the model from silhouettes. However, instead of contours, they use a skeleton based generative shape model. Also, their detection stage is using dynamic programming, which is quite different from our method. Besides pure shape based method, Maji and Malik [19] propose a maximum margin hough voting method with SVM to detect objects. Gu et al. [11] combine the region and shape together for object detection.

Particle filter (PF) has been used for object detection previously [13,12]. They mainly utilize PF to reduce the possible assumptions and they have pre-defined the order for PF. However, our method can determine the order of PF on the fly, which is theoretically quite different from the traditional PF. Moreover, we are based on shape features for object detection instead of the binary classifier they defined. Lu et al. [18] also use PF for shape based object detection. However, we are totally different from them at the proposal and evaluation steps, which is essential for PF. Also, the pairwise relation between parts is naturally embedded into our PF framework, which has not been done in the previous PF methods.

### 3 Partially-Supervised Model Learning

Our approach only requires marking object parts on one exemplar. We then transfer this knowledge to other contours not only in the same shape class but also to similar shape classes. Thus, our approach is able to construct the part models for different classes of objects starting with only one exemplar contour. The constructed model can describe a wide range of objects with different poses.

As we learn the model from exemplars, the first issue is which ones should be chosen from a given training data set. We use Affinity Propagation to select the exemplars, which are cluster centers in AP. These cluster centers are representative, so that they can describe most of the poses of objects. The input pairwise distance between shapes is obtained by Oriented Chamfer Matching (OCM).

#### 3.1 Part Model Construction

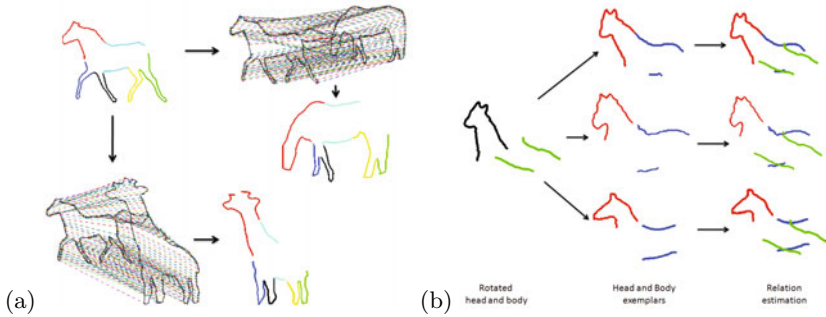
In this section we describe a way to automatically decompose the exemplars  $E = \{E_1, \dots, E_{N_e}\}$  into meaningful parts. We first manually segment one selected silhouette, say  $E_1$  into  $m$  different meaningful parts  $S = \{s_1, \dots, s_m\}$ . For example, for horse, we have six parts: head, two front legs, two back legs, and the body, shown in different colors in top left of Fig. 2(a). We then use shape matching with IDSC [17] to transfer the parts to other exemplars  $E_2, \dots, E_{N_e}$ , e.g., to the second horse in Fig. 2(a). The corresponding points carry over the part decomposition. To ensure that the part decomposition is transferred correctly, we require that the number of corresponding points for a given contour part  $s_i$  is larger than a given threshold, e.g. 80% of the total number of points in the contour part. If this is not the case, the corresponding part is removed from the model.

We define part bundle  $B_i$  as a set composed of part  $s_i$  on  $E_1$  and all corresponding parts on  $E_2, \dots, E_{N_e}$  transferred by the IDSC matching for  $i = 1, \dots, m$ . Each part bundle  $B_i$  has at most  $N_e$  contour parts. We obtain a set of  $m$  part bundles  $B = \{B_1, B_2, \dots, B_m\}$  that defines the nodes of our part model graph.

We can also employ shape matching to transfer the part structure to different but similar object classes. As illustrated in Fig. 2(a), our part decomposition of the horse contour transfers easily to contours of giraffes. As long as the objects in different classes have similar structure, the proposed approach can transfer the structure knowledge from the known class to the other classes and obtain the part bundle models. There are three advantages of the proposed approach: 1) It requires very little manual labeling. 2) The constructed model composed of part bundles can handle the intra-class variations as long as the training silhouettes can represent the possible poses of objects. 3) The structural knowledge can be easily transferred to different classes.

#### 3.2 Relation between Model Parts

After learning the model from silhouettes, in order to make the model more flexible, we permit the rotation for each part and also some shift. However, with



**Fig. 2.** (a) Six manually labeled parts on the horse in top left are marked with different colors. The point correspondence obtained by shape matching allows us to transfer the part structure to a different horse and to a giraffe. (b) The horse head and horse body shown on the left hand side are very different from our perception of a horse. Our measure of this fact is illustrated in the rest of this figure.

the increasing flexibility, the obtained model can be very different from shapes in a given object class. To reduce the negative effect of flexible models, we propose a soft way to constrain the flexibility. We allow the flexibility in a range determined by shape similarity to example shapes in a given object class. Here the shape similarity is described by spatial layout of model parts, i.e., a new rotated spatial layout of parts is allowed if it is similar to a layout previously seen for this class. An example is shown in Fig. 2(b). The horse head and horse body shown on the left hand side are very different from our perception of a horse. The head and body are too far away from each other and their arrangement due to rotation is really strange. With the method described below, we can offer a soft constraint on possible spatial layout of parts.

The key idea is to construct a distribution describing the spatial layout between different parts. In particular, given a part bundle  $B_i$ , the spatial relation between it and another part bundle  $B_j$  forms a distribution. This kind of distribution has been used in object detection to help describe the model [23, 5], but the distribution is assumed to be Gaussian, whose parameters can be easily learned from training samples. However, obviously, the distribution of part relation is very complex and expressing it as Gaussian or any other parametric distribution does not seem to be a good approximation. Instead, we propose to learn the underlying distribution in a non-parametric setting.

We employ kernel density estimation, which is one of the most popular non-parametric methods. Given are two rotated parts  $p'_i$  and  $p'_j$  that come from different part bundles  $B_i$  and  $B_j$  respectively. Our goal is to find how is  $p'_j$  located with respect to  $p'_i$ . For example, we want to find out how well the green body is positioned with respect to the black horse head in Fig. 2(b). For part  $p'_j$ , we use  $OCM_{p'_i}$  to find the top  $k$  most similar exemplar parts  $(p_i(1), \dots, p_i(k))$  in part bundle  $B_i$  (the bundle of  $p'_i$ ). For these original parts in  $B_i$ , we know the exemplar contours they came from. From these contours, we extract parts

$(p_j(1), \dots, p_j(k))$  that belong to the same bundle as  $p'_j$ , i.e., to part bundle  $B_j$ . In Fig. 2(b), OCM retrieves the 3 red horse heads  $(p_i(1), p_i(2), p_i(3))$  as most similar to the black head, which in turn carry over from their original contours 3 blue horse bodies  $(p_j(1), p_j(2), p_j(3))$ . Finally, we measure the spatial layout between parts  $p'_i$  and  $p'_j$  by estimating the fitness of  $p'_j$  to the distribution described by  $(p_j(1), \dots, p_j(k))$ :

$$f(p'_j|p'_i) = \frac{1}{C_c} \sum_{t=1}^k \frac{1}{h} K\left(\frac{OCM_{p'_j}(p_j(t))}{h}\right) \quad (1)$$

where  $K$  is a kernel function with bandwidth  $h$ , which is Gaussian in the paper and  $C_c$  is a constant value. The computation of  $f(p'_j|p'_i)$  in our example is illustrated in the right column of Fig. 2(b). It is a function of the OCM distance between the green horse body and the 3 blue horse bodies.

## 4 Framework for Object Detection

Our goal is to infer the maximum of a posterior distribution  $p(B_1, \dots, B_m | Z)$ , where  $(B_1, \dots, B_m)$  is a vector of random variables (RVs) representing part bundles, which are nodes of our shape model graph (3). In our application  $Z = (I, C)$  is a set of observations, where  $I$  is a RV ranging over binary edge images and  $C$  ranges over classes of target objects including background. Thus,  $Z$  is static, since the target edge image and the class of object are fixed for a given detection process. The possible values of each RV  $B_i$  are vectors of two elements, one is the location  $x_i$  in the image and the second is the part  $s_i$  chosen from the part bundle  $B_i$  in the model. In the case of a correct detection, we expect part  $s_i$  to be located at  $x_i$  in the image. We stress that even though each part bundle has many parts, only one of them is chosen for a given location in the image. To simplify the notation, we use  $b$  to represent the pair of values  $(x, s)$  for each random variable, i.e.,  $b_t = (x_t, s_t)$ . Consequently, our goal is to find value assignments to RVs  $B_t = b_t$  for  $t = 1, \dots, m$  that maximize the posterior

$$\hat{b}_{1:m} = \underset{b_{1:m}}{\operatorname{argmax}} p(b_{1:m} | Z), \quad (2)$$

where  $b_{1:m}$  is a shorthand notation for  $(b_1, \dots, b_m)$ . We will achieve our goal by approximating the posterior distribution with a finite number of particles in the framework of Particle Filter (PF). Besides, only a small subset of the search space is considered in the framework, which reduces the complexity significantly compared to exhaustive search with sliding windows, e.g., [22].

Unlike the standard PF framework, the observations  $Z$  in our approach do not arrive sequentially, but are available at once, i.e.,  $Z$  is static. Therefore, the observations have no natural order. Consequently, the states  $b_{1:m}$  also do not have any natural order, i.e., the order of indices  $1, \dots, m$  does not have any particular meaning. Therefore, we need to extend the PF framework to infer an order of RVs, which may be different for each particle. Intuitively, we want to

determine such an order of RVs so that the corresponding order of observations is most informative, which makes the particle reaches optimal solution faster and more accurate. This makes the proposed PF fundamentally different from classical PF. To represent the order of RVs we need a symbol of a bijection (onto and one-to-one function)  $\langle \cdot \rangle^{(i)}: \{1, \dots, m\} \rightarrow \{1, \dots, m\}$ . Although we may have a different bijection for each particle ( $i$ ), we will drop the index ( $i$ ) from  $\langle 1 : t \rangle^{(i)}$ , since the state variables already carry the particle index. For example, we denote  $(b_4^{(i)}, b_5^{(i)}, b_2^{(i)})$  as  $b_{\langle 1:3 \rangle}^{(i)}$ , where  $\langle 1 : 3 \rangle = (4, 5, 2)$ .

We first present the proposed PF algorithm followed by a discussion of its major differences to standard PF approaches. As it is often the case in PF applications, we assume the proposal distribution to be  $q(b|b_{\langle 1:t-1 \rangle}^{(i)}, Z) = p(b|b_{\langle 1:t-1 \rangle}^{(i)})$ . For each particle ( $i$ ), where  $i = 1, \dots, N$ , the proposed PF algorithm in each iteration  $t = 2, \dots, m$  performs the following three steps:

1) **Importance sampling / proposal:** Sample followers of particle ( $i$ ) for  $l \in \{1, \dots, m\} \setminus \langle 1 : t - 1 \rangle$

$$b_l^{(i)} \sim p(b_l|b_{\langle 1:t-1 \rangle}^{(i)}) \tag{3}$$

and set  $b_{\langle 1:t-1 \rangle, l}^{(i)} = (b_{\langle 1:t-1 \rangle}^{(i)}, b_l^{(i)})$ . In particular, in the first iteration ( $t = 1$ ) we generate samples from each dimension of the state space, i.e., we sample for  $l \in \{1, \dots, m\}$

$$b_{\langle 1 \rangle}^{(i)} = b_l^{(i)} \sim p(b_l) \tag{4}$$

2) **Importance weighting/evaluation:** An individual importance weight is assigned to each follower of each particle by

$$w(b_{\langle 1:t-1 \rangle, l}^{(i)}) = p(Z|b_{\langle 1:t-1 \rangle, l}^{(i)}) \tag{5}$$

3) **Resampling:** At the sampling step we have generated more samples than the number of particles. Thus we have a larger set of particles  $b_{\langle 1:t-1 \rangle, l}^{(i)}$  for  $i = 1, \dots, N$  and  $l \in \{1, \dots, m\} \setminus \langle 1 : t - 1 \rangle$  from which we sub-sample  $N$  particles and assign equal weights to all of them as in the standard Sampling Importance Resampling (SIR) approach. We obtain a set of new particles  $b_{\langle 1:t \rangle}^{(i)}$  for  $i = 1, \dots, N$ . The resampling is not performed in the last step, i.e., when  $t = m$ .

**Algorithm discussion:**

1) This step provides our main extension of the classical PF framework. In the classical PF framework, followers of each particle are selected from only one conditional distribution, i.e., from the conditional distribution of RV at dimension  $t$  given by  $p(b_t|b_{1:t-1}^{(i)})$ , since the dimension index  $t$  represents a real order of RVs  $1 : t = 1, \dots, t$ . In contrast we sample the followers from each dimension  $l \in \{1, \dots, m\}$  that is not already included in  $\langle 1 : t - 1 \rangle$ .

The fact that one can consider more than one follower of each particle and reduce the number of followers by resampling is known in the PF literature and



is referred to as prior boosting [10]. It is used to capture multi-modal likelihood regions. However, all followers are selected from the conditional distribution of the same RV (the same dimension  $t$ ) in the classical PF framework.

2) We take the weight formula from [18], where it has been derived for PF with static observations.

3) We stress that the resampling plays in our framework an additional and a very crucial role. It selects the the most informative random variables (i.e., state space dimensions) as followers of particles. Since the weight of  $b_{\langle 1:t-1 \rangle, l}^{(i)}$  is determined by the observations  $Z$ , and the resampling uses the weights to selects a follower  $b_{\langle t \rangle} = b_l$  from not yet considered dimensions  $l \in \{1, \dots, m\} \setminus \langle 1:t-1 \rangle$ , the resampling determines the order of RVs, i.e., the bijection  $\langle t \rangle$  for  $t = 1, \dots, m$ . Consequently, the order of RVs is heavily determined by  $Z$ , and this order may be different for each particle ( $i$ ). This is in strong contrast to the classical PF, where observations  $Z$  have no influence on the order of RVs, which is fixed.

In order to execute the derived PF algorithm, we need to define the proposal distribution  $p(b_l | b_{\langle 1:t-1 \rangle}^{(i)})$ , and the evaluation pdf  $p(Z | b_{\langle 1:t-1 \rangle, l}^{(i)})$ . As stated in Eq. 4 the initial proposal distribution is defined by  $p(b_l)$ , where  $l$  is an index of a RV representing a part bundle and  $b_l = (s_l, x_l)$ . In our implementation,  $p(b_l)$  is simply the probability of finding model part  $s_l$  at location  $x_l$ , and it measures how well model part  $s_l$  fits the edges in the image. We compute it as a Gaussian of the oriented chamfer distance. Similarly,  $p(b_l | b_{\langle 1:t-1 \rangle}^{(i)})$  is the probability of finding model part  $s_l$  at the location  $x_l$ , but now the location is constrained, since parts  $s_{\langle 1:t-1 \rangle}$  have already been placed in the image. Thus, this conditional probability is picked around the expected location  $x_l$  determined by the locations  $x_{\langle 1:t-1 \rangle}$  of the previously added parts. While the initial proposal distribution is computed at every image location, the conditional proposal distribution is only computed at regions of interest determined by the previously placed model parts.

As  $Z = (I, C)$ , and  $I$  and  $C$  can be viewed as independent conditioned on  $b_{\langle 1:t-1 \rangle, l}^{(i)}$ , we obtain:

$$p(Z | b_{\langle 1:t-1 \rangle, l}^{(i)}) = p(I | b_{\langle 1:t-1 \rangle, l}^{(i)}) p(C | b_{\langle 1:t-1 \rangle, l}^{(i)}) \tag{6}$$

We recall that in our detection framework, both  $I$  and  $C$  are instantiated, since they are given prior to the detection, i.e.,  $I = im$ , where  $im$  is a given binary edge image and  $C = 1$ , which represents the class of the target object. The first factor  $p(I = im | b_{\langle 1:t-1 \rangle, l}^{(i)})$  in Eq. 6 describes the goodness of fit to the edge image  $im$  of the partial shape model determined by  $b_{\langle 1:t-1 \rangle, l}^{(i)}$ , i.e., how likely the edges in  $im$  come from a picture of a shape like the shape of  $b_{\langle 1:t-1 \rangle, l}^{(i)}$ . The second factor  $p(C = 1 | b_{\langle 1:t-1 \rangle, l}^{(i)})$  represents the probability of the target class given the model  $b_{\langle 1:t-1 \rangle, l}^{(i)}$ . Hence it can be viewed as shape class constraints on the model. The conditional pdfs describing both factors are defined in § 5.

## 5 Evaluation Based on Shape Similarity

As  $b_{<1:t-1>,l}^{(i)}$  consists of the parts  $s_{<1:t-1>,l}^{(i)}$  and their locations  $x_{<1:t-1>,l}^{(i)}$ , we construct a partial shape model  $\mu$  by putting parts  $s_{<1:t-1>,l}^{(i)}$  at locations  $x_{<1:t-1>,l}^{(i)}$  on the edge map  $im$ . The probability that the edge map  $im$  is an image of a real object looking like our partial model  $\mu$  is given by

$$p(I = im | b_{<1:t-1>,l}^{(i)}) = \exp(-\beta \cdot OCM_{im}(\mu)), \quad (7)$$

where  $OCM_{im}(\mu)$  returns the Oriented Chamfer distance between  $im$  and  $\mu$  and  $\beta$  is set to 10. Consequently,  $OCM_{im}(\mu)$  measures how well the constructed partial model matches to the edge map.

$p(C = 1 | b_{<1:t-1>,l}^{(i)})$  expresses the probability of the target shape class given partial shape model  $\mu = b_{<1:t-1>,l}^{(i)}$ . We obtain by Bayes rule

$$p(C = 1 | \mu) = \frac{p(\mu | C = 1)p(C = 1)}{\sum_{c=1,0} p(\mu | C = c)p(C = c)}. \quad (8)$$

$p(\mu | C = 1)$  measures the similarity between the constructed model and the target class. Similarly,  $p(\mu | C = 0)$  measures the similarity between the constructed model and the background. Eq. 8 helps to prevent accidental match to the background, since it eliminates shape models with both high similarity to a given object class and to the background, and favors models with high similarity to a given object class and low similarity to the background. We utilize a recursive computation in our PF framework to obtain

$$\begin{aligned} p(\mu | C = c) &= p(b_{<1:t-1>,l}^{(i)} | C = c) \\ &= p(b_l^{(i)} | b_{<1:t-1>,l}^{(i)}, C = c) p(b_{<1:t-1>,l}^{(i)} | C = c) \\ &= p(b_l^{(i)} | b_{<t-1>,l}^{(i)}, C = c) p(b_{<1:t-1>,l}^{(i)} | C = c) \\ &= f(b_l^{(i)} | b_{<t-1>,l}^{(i)}) p(b_{<1:t-1>,l}^{(i)} | C = c), \end{aligned} \quad (9)$$

where  $f$  is defined in Eq. 4 and a given shape class  $C = c$  is modeled as a set of exemplars  $E = \{E_1, \dots, E_{N_e}\}$ , which are selected from training examples by affinity propagation.  $f$  describes the pairwise relation between nodes in the graph, which is naturally utilized in our PF framework. When  $C = 0$ , we randomly select some background edge configurations as training examples. In the transition from 2nd to 3rd row in Eq. 9, we make a Markov assumption that the new model part  $b_l^{(i)}$  only depends on the previously added part  $b_{<t-1>,l}^{(i)}$  conditioned that we know the shape class  $C = c$ . This simplifies the computation and makes the shape model more flexible in that the pose of the new model part is only evaluated with respect to the pose of previously added part. Finally,  $p(b_{<1:t-1>,l}^{(i)} | C = c)$  is remembered from the previous iteration of particle  $(i)$ .

## 6 Experimental Results

We have tested our algorithm on three widely used data sets: the extended Weizmann Horses [20,22], the ETHZ shapes [7] and the TU Darmstadt Database [16]. During the testing for Weizmann Horses, only 12 automatically selected horse silhouettes with one hand decomposed horse are used to learn the shape model. All the other images are used for testing. The edge maps for this dataset are obtained by Canny edge detector. We also test our method on the class of giraffe in ETHZ shape dataset [7]. The reason why we only select the category giraffes from ETHZ is that our model learning method can only transfer between objects with similar structure and giraffe is the only object in ETHZ having similar structure to horse. Only one hand decomposed horse and 6 automatically selected giraffe silhouettes are used to learn the giraffe model. Further, we work on the cow dataset the TU Darmstadt Database [16], since cows have similar structure with the above two classes. It contains 111 images. Only one hand decomposed horse and 6 automatically selected cow silhouettes are used to learn the cow model. The edge maps for this dataset are obtained by Canny edge detector.

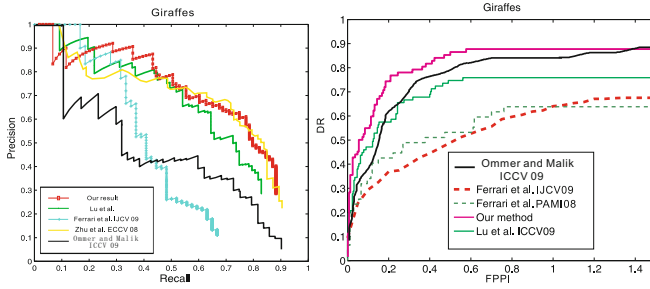
To adapt to large scale variance, we generate multiple models by resizing the original ones to 5 to 8 scales, and choose as the final result from the best score in all the scales. We not only report our results on the commonly used bounding box intersection, but also the accuracy of our boundary localization.

### 6.1 Detection according to Bounding Boxes

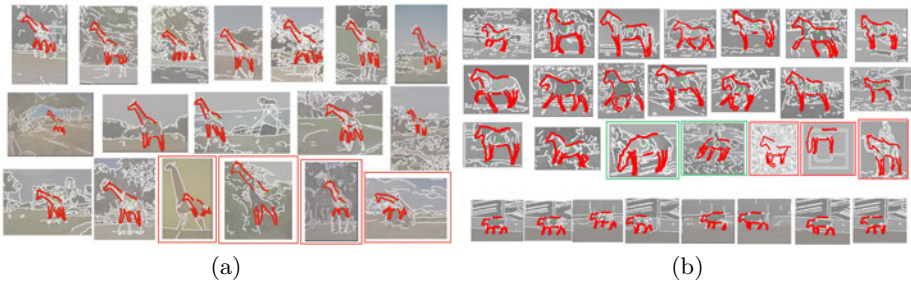
We first evaluate the ability of the proposed approach to localize objects in cluttered images using bounding-box intersection, which is widely used in traditional object detection task. We adopt the strict standards of PASCAL Challenge criterion: a detection is counted as correct only if the intersection-over-union ratio with the ground-truth bounding-box is greater than 50%.

Fig. 3 reports precision-recall (P/R) curve and detection rate vs false positive per image (DR/FPPI) curve for the class Giraffes in ETHZ dataset. In P/R, we compare to Lu et al. [18], Zhu et al. [27], Ommer and Malik [20] and Ferrari et al. [7], whose results are quoted from [18]. In DR/FPPI, as Ferrari et al. [7,6], Ommer and Malik [20] and Lu et al. [18] provide their results, we compare to them. As Ravishankar et al. [21] do not give their curves, we do not compare to them in Fig. 3. According to the curves, we are better than Lu et al. [18], Ommer and Malik [20], Ferrari et al. [7,6] and perform equally well as Zhu et al. [27]. The performance of the proposed method illustrates its ability to cope with substantial nonrigid deformations, which are present in the class Giraffes. This is demonstrated by our example results in Fig. 4(a).

Table 1 compares our detection rate to [26,22] on Weizman Hores and TU Darmstadt Cows. The detection rate on horses is estimated from the DR/FPPI curve in [22]. The DR/FPPI curve for cows is not available in [22]. The method in [26] is also matching based, while [22] is a classification method. Some examples of our horse and cow detection results are shown in Fig. 4(b). The detection precision/recall area under curve (AUC) is a standard performance measure on



**Fig. 3.** Precision-recall curve and detection rate (DR) vs false positive per image (FPPI) curve for the class Giraffes in ETHZ dataset



**Fig. 4.** Examples of detection results for Giraffes, horses and cows

the Weizmann Horses dataset. The AUC for our approach is 79.84%, which is comparable to the result 80.32% in Xiang et al. [11]. We compare to them as they also use the explicit shape model and matching based method for object detection. The AUC of classification based methods [22,9] is 84.98% and 96%, respectively. We observe that classification based methods are bounding box classifiers and utilize significantly more information than matching based methods as ours. This explains why our detection rate and AUC is lower than [22,9].

The proposed approach can not only succeed in extensive cluttered images, but also handles the problem of large range of scales and intra-class variability. This is demonstrated by several examples in Fig. 4. The images in the bottom right of Fig. 4(a) with red rectangles are the ones we fail to detect. The images of horses in Fig. 4(b) with red rectangles are false positives in the negative images provided by Shotton et al. [22] to complement the Weizmann horse dataset.

**Table 1.** Detection rate

	Our method	Zhu et al. [26]	Shotton et al. [22]
Horses	93.97%	86.0%	95.20%
Cows	90.38%	88.6%	N/A

They show that the false positives in the negative set are caused by really very cluttered edges or by the structure of edges happening to match to the model very well. Interestingly, the rightmost false positive of horses is due to a camel, whose shape is very similar to that of a horse.

## 6.2 Localizing Object Boundaries

The method presented in this paper offers one important advantage compared to texture based and classification methods like [3,9,4]. It can localize object boundaries, rather than just bounding-boxes.

In order to quantify how accurately the output shapes match to true boundaries, we use the coverage and precision measures defined in [7]. Coverage is the percentage of points from ground-truth boundaries closer than a threshold  $t$  to the output shapes of the proposed approach. Reversely, precision is the percentage of points from output shapes closer than  $t$  to any point of ground-truth boundaries. As in [7]  $t$  is set to 4% of the diagonal of the ground-truth bounding box. The measures are complementary. Coverage captures how much of the object boundary has been recovered by the algorithm, whereas precision reports how much of the algorithm's output lies on the object boundaries. These measurements are really useful and suitable for evaluating shape based approaches. In comparison, bounding-box evaluation cannot represent how accurate the detected shapes match the ground-truth boundary. It is possible to have bounding-box intersection larger than 0.5 without having correctly identified the ground-truth object boundaries. Two examples of horse detection are shown in Fig. 4(b) with green rectangles.

The first two columns of Table 2 show coverage and precision averaged over all images of the class giraffes in ETHZ dataset in comparison to the results in [7]. We measure the coverage and precision for the correct detections at 0.4 FPPI, following [7]. The coverage of the proposed approach is over 11% better than [7], which shows that our approach can efficiently recover the true boundary of objects. The precision is a little lower than [7]. More importantly, the detection rate at our 0.4 FPPI is 86.75%. However, even for 20% bounding box intersection, the detection rate at 0.4 FPPI in [7] is only around 60%, which is much less than us. It demonstrates that our approach can correctly localize object's boundary on more images.

For horses and cows, the coverage and precision are obtained over all correct detections. The third column of Table. 2 shows the coverage and precision of the proposed method on the Weizmann Horse dataset. As the edges are significantly worse than the ones provided for the giraffes, both measures are worse than the results on giraffes. The coverage and precision results for cow are shown in the fourth column of Table. 2. Due to less intra-shape variance, the precision is 92.02%, which is much higher than giraffes and horses. However, the coverage is only 73.86%. The main reason for the difference between these two values is that our model has a gap, since we removed the contour part representing the horse tail from the horse contour used for part decomposition. Thus, even if the model and object match perfectly, the coverage score cannot be perfect (see examples in Fig. 4).

**Table 2.** Accuracy of the boundary localization

	Ours on giraffes	Results in [7] on giraffes	Ours on horses	Ours on cows
Coverage	79.4%	68.5%	77.5%	73.86%
Precision	74.6%	77.3%	61.7%	92.02%

## 7 Conclusion and Discussion

This paper mainly contains two contributions: shape model learning through shape matching and a novel framework for shape based object detection. The proposed model learning method can not only learn the model for non-rigid or articulated objects with partially-supervised learning, but also transfer the structure information to different kinds of objects. More importantly, the spatial layout between parts is also modeled.

We extend the classical particle filter framework in order to be able to infer an optimal label assignment to RVs whose dependencies are described by a complete graph. The values of RVs represent contour parts of our shape model and their locations. In our framework each particle explores a different order of detected contour parts, and the most informative order is selected by particle resampling.

## Acknowledgments

The work has been supported by the NSF Grants IIS-0812118, BCS-0924164, the AFOSR Grant FA9550-09-1-0207, and the DOE Award 71498-001-09.

## References

1. Bai, X., Wang, X., Latecki, L.J., Tu, Z.: Active skeleton for non-rigid object detection. In: ICCV (2009)
2. Borenstein, E., Sharon, E., Ullman, S.: Combining top-down and bottom-up segmentation. In: POVC (2004)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
4. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for multi-class object layout. In: ICCV (2009)
5. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. IJCV 61(1) (2005)
6. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: From images to shape models for object detection. IEEE Trans. PAMI 30(1), 36–51 (2008)
7. Ferrari, V., Jurie, F., Schmid, C.: From images to shape models for object detection. IJCV 87(3) (2010)
8. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science 315(5814), 972–976 (2007)
9. Gall, J., Lempitsky, V.: Class-specific hough forests for object detection. In: CVPR (2009)

10. Gordon, N., Salmond, D., Smith, A.: Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings of Radar and Signal Processing* 140, 107–113 (1993)
11. Gu, C., Lim, J.J., Arbelaez, P., Malik, J.: Recognition using regions. In: *CVPR* (2009)
12. Ioffe, S., Forsyth, D.: Finding people by sampling. In: *ICCV* (1999)
13. Ioffe, S., Forsyth, D.: Probabilistic methods for finding people. *IJCV* (2001)
14. Kokkinos, I., Yuille, A.: Hop: Hierarchical object parsing. In: *CVPR* (2009)
15. Lee, Y.J., Grauman, K.: Shape discovery from unlabeled image collections. In: *CVPR* (2009)
16. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: *Proceedings of the Workshop on Statistical Learning in Computer Vision, Prague, Czech Republic* (May 2004)
17. Ling, H., Jacobs, D.: Shape classification using the inner-distance. *IEEE Trans. PAMI* 29, 286–299 (2007)
18. Lu, C., Latecki, L.J., Adluru, N., Yang, X., Ling, H.: Shape guided contour grouping with particle filters. In: *ICCV* (2009)
19. Maji, S., Malik, J.: Object detection using a max-margin hough transform. In: *CVPR* (2009)
20. Ommer, B., Malik, J.: Multi-scale object detection by clustering lines. In: *ICCV* (2009)
21. Ravishankar, S., Jain, A., Mittal, A.: Multi-stage contour based detection of deformable objects. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part I*. LNCS, vol. 5302, pp. 483–496. Springer, Heidelberg (2008)
22. Shotton, J., Blake, A., Cipolla, R.: Multi-scale categorical object recognition using contour fragments. *IEEE Trans. on PAMI* 30(7), 1270–1281 (2008)
23. Stark, M., Goesele, M., Schiele, B.: A shape-based object class model for knowledge transfer. In: *ICCV* (2009)
24. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: *CVPR* (2001)
25. Yuille, A., Coughlan, J., Wu, Y., Zhu, S.: Order parameters for detecting target curves in images, when does high-level knowledge help? *IJCV* 41(1/2), 9–33 (2001)
26. Zhu, L., Chen, Y., Yuille, A.: Learning a hierarchical deformable template for rapid deformable object parsing. *IEEE Trans. PAMI* 99(1) (2009)
27. Zhu, Q., Wang, L., Wu, Y., Shi, J.: Contour context selection for object detection: a set-to-set contour matching approach. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part II*. LNCS, vol. 5303, pp. 774–787. Springer, Heidelberg (2008)

# Geodesic Shape Retrieval via Optimal Mass Transport<sup>\*</sup>

Julien Rabin, Gabriel Peyré, and Laurent D. Cohen

CEREMADE, Université Paris-Dauphine  
{rabin,peyre,cohen}@ceremade.dauphine.fr

**Abstract.** This paper presents a new method for 2-D and 3-D shape retrieval based on geodesic signatures. These signatures are high dimensional statistical distributions computed by extracting several features from the set of geodesic distance maps to each point. The resulting high dimensional distributions are matched to perform retrieval using a fast approximate Wasserstein metric. This allows to propose a unifying framework for the compact description of planar shapes and 3-D surfaces.

## 1 Introduction

Content based 2-D and 3-D shape retrieval is an important problem in computer vision. It requires to design both representations and similarity measures to discriminate shapes from different classes, while being invariant to some deformations.

### 1.1 Feature-Based Shape Retrieval

There is a large amount of literature on content-based retrieval using similarity measures between descriptors. In this section, a brief review is given, focusing on bending and isometric deformations (*i.e.* preserving the topology). We refer the reader to the following review papers devoted to planar shapes [12] and 3-D surfaces [3,4] retrieval for a complete review.

*Global descriptors.* Simple global features are computed using polynomial moments [5,6,7], or Fourier transform [8] (see [9] for review).

The spectrum of the Laplace Beltrami operator defines a descriptor invariant to rigid motion and to simple bendings [10]. Shape distributions [11] compute descriptors as histograms of the distribution of Euclidean distance between points on the surface. This is extended to bending invariant descriptors in [12,13,14] using geodesic distances. It is possible to replace the geodesic distance by a diffusion distance [15] computed by solving a linear Poisson PDE.

---

<sup>\*</sup> This work has been done with the support of the French “Agence Nationale de la Recherche” (ANR), under grant NatImages (ANR-08-EMER-009).



*Local descriptors.* Many other shape representations do not make use of a single descriptor. They rather compute similarities by matching points of interest for which local descriptors are defined. Shape context features [16] are local 2-D histograms of contours around points of interest. Geodesic shape context makes use of geodesic curves to gain bending invariance [17]. Local tomographic projections on tangent planes (spin images) [18] define a set of local descriptors.

*Similarity measure.* Most of the previous approaches make use of Euclidean metric, Kullback-Leibler or  $\chi^2$  distance to compare low-dimensional histogram-based descriptors in linear time. When considering high-dimensional descriptors (either histograms or discrete point clouds), another possibility is to use the Wasserstein distance [19], see *e.g.* [20,21,22].

## 1.2 Contributions

This paper introduces a novel framework for bending invariant recognition of shapes. We use the setting of geodesic distances on Riemannian manifolds, which unifies both planar shape and 3-D surface retrieval problems. This novel framework builds on several already known statistical descriptors, and encompasses them into a single high-dimensional descriptor. This allows us to take advantage of the richness of information available in each separate statistical measure to enhance the retrieval performance. The retrieval method is based on an approximation of the Wasserstein distance, that works directly over discrete point clouds, and can be computed with an iterative algorithm.

## 2 Geodesic Distances

In the following, we consider shapes as compact 2-D manifolds  $\Omega \subset \mathbb{R}^s$ , where  $s = 2$  (planar shapes) or  $s = 3$  (surfaces). Note however that our approach is generic and accommodates for domains of arbitrary dimension.

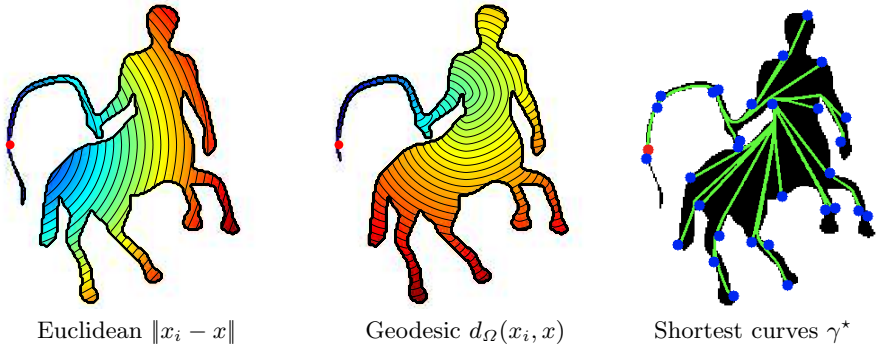
### 2.1 Geodesic Distance Definition

The length of a curve  $\gamma : [0, 1] \rightarrow \Omega$  traced within the domain is defined as  $L(\gamma) = \int_0^1 \|\gamma'(t)\| dt$ . The geodesic distance between two points  $x_s, x_e \in \Omega$  is the length of the shortest piecewise smooth curve joining the two points

$$d_\Omega(x_s, x_e) = \min_{\gamma(0)=x_s, \gamma(1)=x_e} L(\gamma). \quad (1)$$

The geodesic map  $d_\Omega(x_i, x)$  differs significantly from the the Euclidean distance map  $\|x_i - x\|$  when the shapes are non convex, as it is illustrated by Fig. 1.

A curve  $\gamma^*$  satisfying  $d_\Omega(x_s, x_e) = L(\gamma^*)$  is called a shortest path, sometimes also referred to as a (globally minimizing) *geodesic*. Figure 1 (on the far right) shows several examples of geodesics, each time computed between a starting point  $x_s \in \Omega$  (red dot) and some ending points  $\{x_e\}$  lying on the boundary  $\partial\Omega$  of the manifold (blue dots).



**Fig. 1.** Left and center: comparison of Euclidean and geodesic distances inside a 2-D shape. Right: display of geodesic curves.

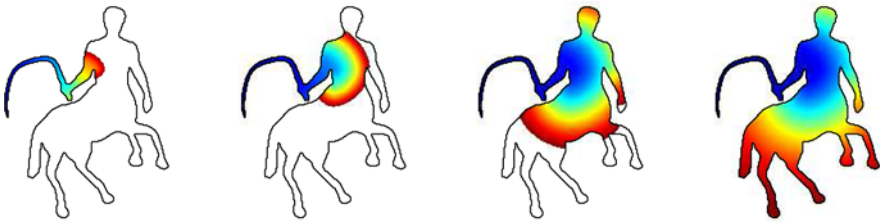
**2.2 Geodesic Distance Computation**

*Geodesic distance within a planar shape.* Given some starting point  $x_s$ , the geodesic distance map  $U_{x_s}(x) = d_\Omega(x_s, x)$  can be shown to be the unique viscosity solution of the following non-linear PDE,

$$\forall x \in \Omega, \quad \|\nabla U_{x_s}(x)\| = 1 \quad \text{and} \quad U_{x_s}(x_s) = 0. \tag{2}$$

where the derivative should be understood in a weak sense at points along the medial axis of  $x_s$  where  $U_{x_s}$  is not smooth.

The PDE (2) can be discretized with upwind finite difference. The resulting discrete equation can be and solved in  $O(N \log(N))$  operations using the Fast Marching algorithm [23,24]. This algorithm performs a front propagation within the shape, as displayed on Fig. 2.



**Fig. 2.** Fast Marching propagation inside a 2-D shape

*Geodesic distance on a 3-D surface.* If the surface  $\Omega$  is parametrized using  $\psi : V \subset [0, 1]^2 \mapsto \Omega$ , then one can prove that the distance map

$$\forall x \in V, \quad U_{x_s}(x) = d_\Omega(x_s, \psi(x))$$

satisfies an anisotropic Eikonal PDE

$$\forall x \in V, \quad \|\nabla U_{x_s}(x)\|_{T_x^{-1}} = 1, \quad \text{and} \quad U_{x_s}(\psi(x_s)) = 0, \tag{3}$$

$$\text{where } T_x = \left( \left\langle \frac{\partial \psi}{\partial x_i}, \frac{\partial \psi}{\partial x_j} \right\rangle \right)_{0 \leq i, j \leq 1} \quad \text{and} \quad \|x\|_A = \sum_{0 \leq i, j \leq 1} A_{i,j} x_i x_j.$$

This equation (3) extends to surfaces of arbitrary topology using several charts that parametrize locally the surface.

The Eikonal equation (3) can be discretized on 3-D meshes. In the case of mesh with no obtuse angle, the discrete equation can be solved in  $O(N \log(N))$  operations [25]. For general meshes, the resolution requires more advanced schemes, see for instance [26]. An example of a Fast Marching propagation from a set of starting points on a 3-D shape is given on Fig. 3.

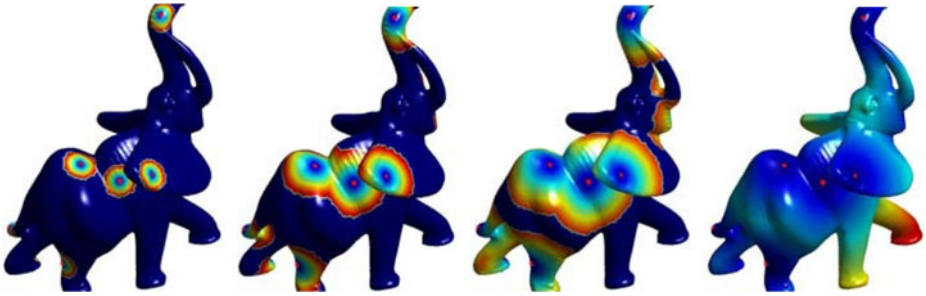


Fig. 3. Example of Fast Marching propagation on a triangulated mesh

### 3 Geodesic Descriptors

Similarity measures between shapes are computed by extracting global or local features  $\varphi(\Omega)$ , and then performing some comparison between the resulting descriptors.

An important goal in designing a similarity measure is to achieve invariance to some class  $\mathcal{R}$  of deformations. This requires that the descriptors are invariant, so that  $\varphi(R\Omega) = \varphi(\Omega)$  for any  $R \in \mathcal{R}$ .

This section details a class of geodesic descriptors that are invariant under geodesic isometries, and quasi-invariant to shape articulations and bendings. This is especially relevant to perform robust retrieval on articulated shapes, such as animal or human with varying poses.

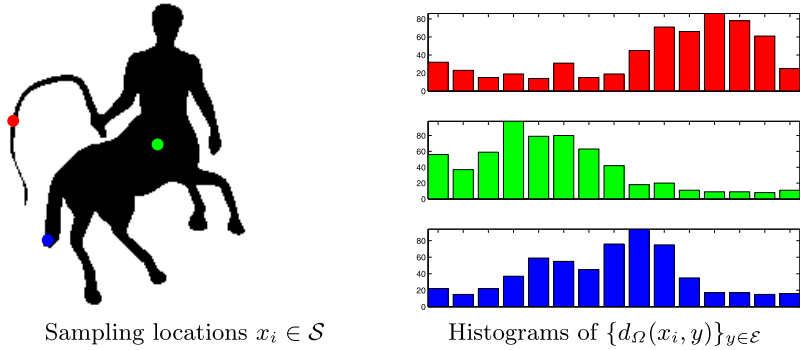
#### 3.1 Local Descriptors

*Geodesic distance distributions.* To design features invariant to bendings and articulations, we consider, for each point  $x \in \mathcal{S} \subset \Omega$ , the set  $\{d_\Omega(x, y)\}_{y \in \mathcal{E} \subset \Omega} \subset \mathbb{R}^+$  of distances to a subset  $\mathcal{E} \subset \Omega$ . This set of distances should be thought as being a 1-D distribution of values in  $\mathbb{R}^+$ .

For numerical applications, the set  $\mathcal{S}$  is a discrete sub-sampling of the manifold computed as described in Sect. 3.2. The set  $\mathcal{E}$  used to compute the distributions can be defined depending on the application. In our numerical examples, we

choose  $\mathcal{E} = \partial\Omega$  to be the boundary of the manifold for 2-D shapes, and  $\mathcal{E} = \Omega$  for 3-D surfaces.

Figure 4 shows examples of geodesic distance distribution, conveniently displayed using 1-D histograms.



**Fig. 4.** Histogram of the distribution of the geodesic distance to several points

*Geodesic quantile measures.* The whole set of distances  $\{d_\Omega(x, y), y \in \mathcal{E}, x \in \mathcal{S}\}$  is too large to be used for retrieval applications. To achieve dimensionality reduction, we retain only a few statistical measures out of this distribution of values. This article considers quantiles statistical measures  $Q_x(\alpha)$  defined as, for all  $\alpha \in [0, 1]$ ,

$$\forall x \in \mathcal{S}, \quad Q_x(\alpha) = F_x^{-1}(\alpha) = \max\{\delta \in \mathbb{R}^+, F_x(\delta) \leq \alpha\} \tag{4}$$

where  $F_x$  is the cumulative distribution function of the set  $\{d_\Omega(x, y), y \in \mathcal{E}\}$ , and  $F_x^{-1}$  is its pseudo-inverse.

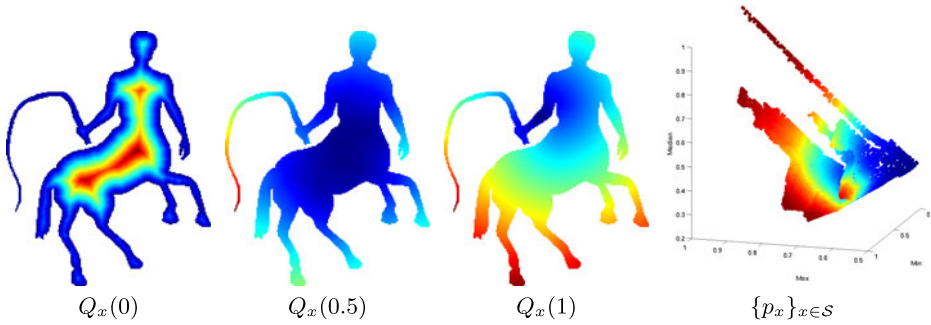
Observe that  $Q_x(0)$  is the minimum geodesic distance between  $x$  and  $\mathcal{E}$ , while  $Q_x(1/2)$  is the median distance. The maximum distance  $Q_x(1)$  is also known as the *eccentricity*, and has been used for 2-D shapes [13] and surfaces [14] retrieval. Other statistical measures can be retained as well. For instance, the mean of the distance  $\int_{\mathcal{E}} d_\Omega(x, y)dy$  is used in [12] to perform surface retrieval.

*Geodesic local descriptors.* At each location  $x \in \mathcal{S}$ , the local descriptor  $p_x \in \mathbb{R}^d$  is a vector of  $d$  quantiles

$$p_x = (Q_x(\alpha_\ell))_{1 \leq \ell \leq d} \in \mathbb{R}^d,$$

where  $0 \leq \alpha_\ell \leq 1$  are equi-spaced values. Figure 5 displays each of the  $d = 3$  components  $Q_x(\alpha_\ell)$  of  $p_x$  as a function of  $x \in \Omega$ , with  $\alpha_\ell \in \{0, \frac{1}{2}, 1\}$ .

The key feature of our approach, that makes it significantly different from these previous works is that it uses several statistical measures, and thus builds high dimensional descriptors.



**Fig. 5.** Display of  $x \mapsto Q_x(\alpha_\ell)$  for several  $\alpha_\ell \in \{0, \frac{1}{2}, 1\}$  and of the corresponding 3-D distribution  $\{p_x = (Q_x(\alpha_\ell))_{1 \leq \ell \leq 3}\}_{x \in S} \subset \mathbb{R}^3$

### 3.2 Global Descriptors

The local descriptors  $p_x$  are sampled on a set  $S \subset \Omega$  to obtain a global descriptor that characterizes the shape.

*Farthest Point Sampling.* Estimating the full set  $\{p_x\}_{x \in \Omega}$  of descriptors is computationally intractable, and one thus needs to compute a sub-sampling  $S = \{x_i\}_{i \in I}$  of  $n$  points on the manifold, where  $I = \{0, \dots, n - 1\}$ . To perform a uniform sampling of the manifold, we use the farthest point sampling strategy. It corresponds to a greedy scheme, originally introduced in [27], and extended to geodesic distances on manifolds for surface remeshing in [28].

The initial point  $x_0 \in \Omega$  is sampled at random. Given a set of  $k$  points  $\{x_0, \dots, x_{k-1}\}$ , the next point is computed as

$$x_k = \operatorname{argmax}_{x \in \Omega} \min_{0 \leq i < k} d_\Omega(x_i, x).$$

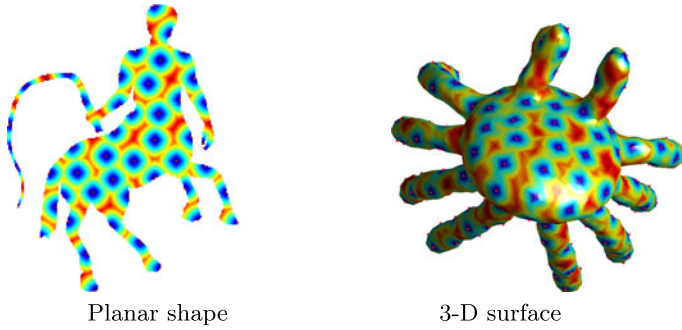
Once this new point  $x_k$  has been computed, the set  $\{d_\Omega(x_k, x)\}_{x \in \Omega}$  of geodesic distances is computed in  $O(n \log(n))$  operations, and the geodesic descriptor  $p_{x_k} \in \mathbb{R}^d$  is obtained by computing the quantiles (4) from these distances. This process of iteratively adding the furthest point to the set  $S$  is continued until a given number of points  $n$  is reached. Examples of this farthest point sampling method on a 2-D shape and a surface are shown in Fig. 6.

*Global descriptor as a point cloud.* The global descriptor is then defined as a uniform sampling of the local descriptors

$$\varphi(\Omega) = \{p_{x_i}\}_{i \in I} \subset \mathbb{R}^d,$$

and is thus a cloud of  $n$  points in  $\mathbb{R}^d$ .

This point cloud  $\varphi(\Omega)$  should be thought as being drawn from a probability distribution. Each shape has its own distribution, that is invariant under isometric bending of the shape.



**Fig. 6. Illustration of the farthest point sampling strategy for a 2-D shape and a 3-D surface.** For each case  $|\mathcal{S}| = n = 100$  points are sampled (red dots). The geodesic distances between each  $x \in \Omega$  and these points are plotted as a colormap.

An alternative representation is to compute the  $d$ -dimensional histogram of the distribution. We prefer in this paper to work directly using discretized point cloud, because it offers a more precise matching.

The resulting global descriptor  $\varphi(\Omega)$  is invariant to isometric deformations, in the sense that  $\varphi(\Omega) = \varphi(R\Omega)$  if  $R$  is a deformation of  $\Omega$  that maintains geodesic distances. More generally, if  $R$  does not modify too much the distances, meaning

$$\forall (x, y) \in \Omega^2, \quad d_\Omega(x, y) \approx d_{R\Omega}(Rx, Ry), \tag{5}$$

then  $\varphi(\Omega) \approx \varphi(R\Omega)$ . This is the case for bending deformations and articulations, see [17]. Observe that geodesic distances  $\{d_\Omega(x, y)\}_{x, y \in \Omega}$  have to be normalized (according to maximum distance) to achieve invariance to scaling.

## 4 Optimal Transport Retrieval

Our shape retrieval method uses a similarity measure that compares the geodesic descriptors  $\varphi(\Omega)$  using the Wasserstein metric related to the Monge-Kantorovich optimal transport problem (see [19] for an in-depth study).

### 4.1 Similarity Measure

In classical settings, shapes are generally represented by histogram-based descriptors and thus compared with  $L^p$  distances or the Kullback-Leibler divergence. In our setting, the descriptors are high dimensional discrete distributions, and we propose to use the Wasserstein distance [19] which is well adapted to compare statistical distributions [20], and is known to be more robust than Hausdorff distance [29]. Our similarity measure is thus defined as

$$\Delta(\Omega_0, \Omega_1) = W(\varphi(\Omega_0), \varphi(\Omega_1)),$$

where  $W(X, Y)$  is the Wasserstein distance between two point clouds  $X, Y \subset \mathbb{R}^d$ , defined in the next section. Since our geodesic descriptors satisfy the approximate invariance (5) for bendings and articulations  $R \in \mathcal{R}$ , our similarity measure satisfies  $\Delta(\Omega_0, R(\Omega_0)) \approx 0$ .

### 4.2 Wasserstein Distance

Given two point clouds  $X, Y \subset \mathbb{R}^d$  of  $n$  elements, the  $L^2$ -Wasserstein distance is defined as

$$W(X, Y)^2 = \min_{\sigma \in \Sigma_n} \sum_{i \in I} \|X_i - Y_{\sigma(i)}\|^2, \tag{6}$$

where  $\Sigma_n$  is the set of all permutations of  $n$  elements and  $I = \{0, \dots, n - 1\}$ . Our framework extends to arbitrary strictly convex cost such as  $L^p$ -metric for  $p > 1$ . Computing this distance boils down to estimate the optimal assignment  $i \mapsto \sigma^*(i)$  minimizing Formula (6). This can be computed exactly using linear programming or other dedicated algorithms in  $O(n^{2.5} \log(n))$  operations [30].

*One-dimensional case.* It is well known that the Wasserstein assignment problem in 1-D can be easily solved in  $O(n \log(n))$  operations by sorting the values [19]. Indeed,  $\sigma_X$  and  $\sigma_Y$  being two permutations such that  $\{X_{\sigma_X(i)}\}_i$  and  $\{Y_{\sigma_Y(i)}\}_i$  are sorted in increasing order, the optimal assignment is

$$\sigma^* = \sigma_Y \circ \sigma_X^{-1}. \tag{7}$$

### 4.3 Approximate Wasserstein Distance

For large point clouds in high dimension ( $d \geq 2$ ), computing exactly (6) is too demanding. Following an idea recently introduced in [31], we propose to use the an approximate transport cost  $\tilde{W}(X, Y)$  defined as

$$\tilde{W}(X, Y) = \|X - X^{(\infty)}\|, \tag{8}$$

where  $X^{(\infty)}$  is computed using an iterative algorithm described in the following paragraph. Starting from  $X^{(0)} = X$ , this algorithm computes points clouds  $\{X^{(k)}\}_k$  that progressively evolves  $X^{(k)}$  toward  $Y$ , minimizing at each iteration an energy  $E_Y$  which is a sum of 1-D Wasserstein distances on the unit sphere  $S^{d-1}$  in  $\mathbb{R}^d$ :

$$E_Y(U) = \frac{1}{2} \int_{\theta \in S^{d-1}} \sum_{i \in I} \langle U_i - Y_{\sigma_\theta^*(i)}, \theta \rangle^2 d\theta, \tag{9}$$

where  $\langle \cdot, \cdot \rangle$  is the  $L^2$ -scalar product, and where  $\sigma_\theta^*$  is the optimal 1-D assignment according to direction  $\theta$  of  $\{Y_i, \theta\}_i$  with  $\{U_i, \theta\}_i$  following Formula (7).

*Algorithm.* Finding a minimum of energy (9) can be done using a classical gradient descent strategy. For numerical considerations, this energy is estimated at each iteration  $k$  from a restricted set of directions, thus resulting in a *stochastic gradient descent* scheme (see e.g. [32]), which relies on three steps:

▷ **Step 1.** Define the direction set  $\Psi^{(k)} \subset S^{d-1}$ , a collection of vectors *randomly and uniformly sampled* on  $S^{d-1}$ . The corresponding energy  $E_Y^{(k)}$  is therefore

$$E_Y^{(k)}(U) = \frac{1}{2} \sum_{\theta \in \Psi^{(k)}} \sum_{i \in I} \langle U_i - Y_{\sigma_\theta^*(i)}, \theta \rangle^2. \tag{10}$$

- ▷ **Step 2.** Compute, for each direction  $\theta \in \Psi^{(k)}$ , the optimal 1-D assignment  $\sigma_\theta^*$  of 1-D distribution  $\{\langle Y_i, \theta \rangle\}_{i \in I}$  with  $\{\langle X_i^{(k)}, \theta \rangle\}_{i \in I}$  using Formula (7).
- ▷ **Step 3.** The set  $\{\sigma_\theta^*\}$  being computed, update the point cloud using a Newton step with parameter  $\lambda \in ]0, 2[$  to minimize the energy  $E_Y^{(k)}$ , such that  $\forall i \in I$

$$\begin{aligned} X_i^{(k+1)} &= X_i^{(k)} - \lambda \left( \nabla^2 E_Y^{(k)}(X^{(k)}) \right)^{-1} \nabla E_Y^{(k)}(X_i^{(k)}), \\ &= (1 - \lambda) \cdot X_i^{(k)} + \lambda H^{-1} \sum_{\theta \in \Psi^{(k)}} \langle Y_{\sigma_\theta^*(i)}, \theta \rangle \theta, \end{aligned} \tag{11}$$

where  $H = \sum_{\theta \in \Psi^{(k)}} \theta \theta^T \in \mathbb{R}^{d \times d}$  is the Hessian matrix of  $E_Y^{(k)}$  at point  $X_i^{(k)}$ .

*Convergence.* Stochastic gradient descent is known to converge if one uses a properly chosen step size  $\lambda = \lambda_k$  that decays through the iterations, see [32]. In numerical simulations, we always observed convergence of  $X^{(k)}$  to some  $X^{(\infty)}$  using the fixed step size  $\lambda = 1/|\Psi^{(k)}|$ . Furthermore,  $X^{(\infty)}$  is actually equal (up to a permutation) to  $Y$ , so that the algorithm computes an assignment between the two distributions –which is not necessarily optimal in the sense of Formula (6).

*Implementation.* In the numerical simulations of Sect. 5, we used a fixed number of  $K = 100$  iterations and  $|\Psi| = d$  directions, and we noticed that using more iterations does not improve significantly the retrieval results. The complexity of the proposed algorithm is therefore  $O(|\Psi|Kn \log(n))$ .

This method extends the algorithm proposed by [33] that makes use of an orthogonal set of direction  $\Psi^k$ , and a descent step size  $\lambda = 1$ . Using a smaller step, e.g.  $\lambda = 1/|\Psi^k|$  is important to ensure the convergence of the method. Using a larger set of directions is useful to obtain a assignment that is closer to the optimal one, see [31] for more details. Observe that other approximation methods has been previously proposed in the literature, e.g. making use of metric embedding [21] or wavelet approximations [34].

## 5 Numerical Examples

Given a database of manifolds  $\{\Omega_j\}_{j \in J}$ , our algorithm for shape retrieval begins by computing the global signature  $\varphi(\Omega_j)$  for each manifold in the database. This is performed in parallel to the farthest point sampling algorithm described in Sect. 3.2. When an input manifold  $\Omega$  is queried in the database, its global signature  $\varphi(\Omega)$  is computed, and the shape in the database are ordered according to the Wasserstein distance approximation  $\tilde{W}(\varphi(\Omega), \varphi(\Omega_j))$ .

To evaluate the retrieval performance of the proposed descriptor, two classical performances curves are displayed:

- the *average recall* curve shows the average number of correct shapes (or “true-positives”) retrieved per query among the  $r$  most similar ones in the dataset. This curve, plotted depending of the rank  $r$ , is obtained by querying every shape in the dataset (the query itself is no used to compute the score);



- the *average precision-recall* curve plots the rate of correct shapes retrieved among  $r$  as a function of the average recall rate.

In order to show the interest of considering high dimensional geodesic statistics, each performance curves are shown for two different descriptors: the aforementioned multi-dimensional descriptor and also a simple 1-D descriptor corresponding to the distribution of *eccentricity* (maximal geodesic distances).

## 5.1 2-D Shape Retrieval

In this setting, the domain  $\mathcal{E}$  of ending points  $\{y_j\}_{j \in J}$  is the boundary  $\partial\Omega$  of the manifold. 4-dimensional distributions of  $n = 500$  points are used as global descriptors, considering 3 quantiles  $Q_x(\alpha_j)$  (minimum, median and maximum) in addition with the mean values of geodesic distances.

We consider first the ‘‘Articulated Shapes’’ dataset of [17] (see Fig. 7(a)), a small dataset being designed to evaluate the robustness of retrieval to bending deformations. Performance curves are shown in Figs. 7(b) and 7(c) for both 4-D and 1-D descriptors. Comparison with state-of-the-art methods [17,22] is also provided in Table 1. One can see that considering several geodesic statistics at the same time enables to catch more sophisticated information on the shape while being more robust to bending deformations. Note that it is not the case when using Euclidean metric with descriptors made of 4 1-D histograms instead of the approximate Wasserstein metric with 4-D discrete distributions.

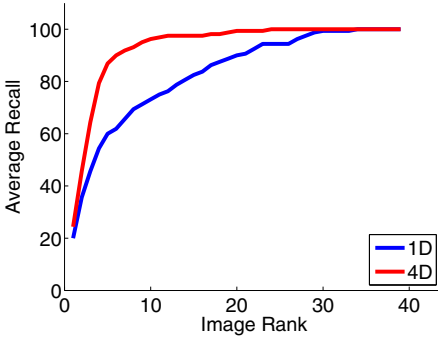
**Table 1. Retrieval results on the articulated shapes dataset [17].** Scores correspond to the number of correct shapes retrieved among 40 depending on their rank.

METHOD		RANK			
DESCRIPTOR	METRIC	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Geodesic quantile distribution (4-D)	Approx. Wasserstein	39	34	30	24
Maximal geodesic distribution (1-D)	Wasserstein	27	24	16	18
4 Geodesic quantile histogram (1-D)	Euclidean	21	15	7	11
Inner Distance Shape Context	$\chi^2$ [17]	40	34	35	27
Inner Distance Shape Context	EMD- $L_1$ [22]	39	39	34	32

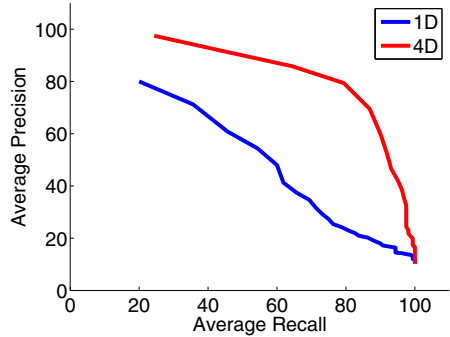
In order to evaluate the robustness of our approach for 2-D shape retrieval on a larger database, we consider now the MPEG-7 dataset of 1400 shapes (see Fig. 8(a)). Results are shown in Figs. 8(b) and 8(c). Again, one can see that using multi-dimensional statistics on geodesic distances yields far better results. The state-of-the art method of [22] yields a bullseye score (average recall rate for rank  $r = 40$ ) of 86.56%, whereas we obtain 59.7%. This can be explained by the strongly non-isometric variations of the objects in the MPEG-7 classes, which makes our representation quite inefficient for this retrieval. An important avenue for future work is to design a large benchmark of planar shapes undergoing bendings and articulation deformations, to explore the performance of our algorithm and related methods.



(a) **Articulated shapes dataset of [17]**. Pairs of shapes from different classes. The complete dataset is composed of 8 classes of 5 elements.



(b) Recall vs Image Rank.

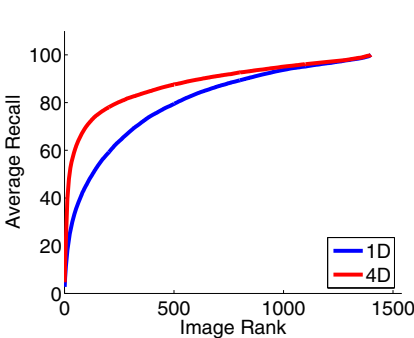


(c) Average Precision-Recall.

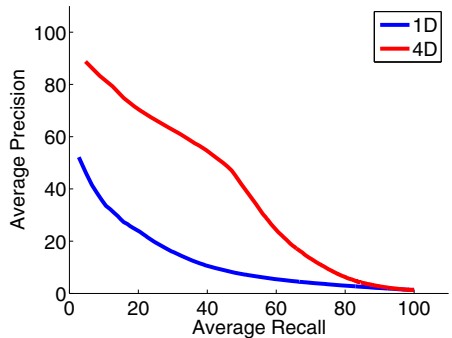
**Fig. 7. Retrieval results on the articulated shapes database [17].** Figure 7(a): database overview. Figure 7(b): Average recall rate depending on the shape rank threshold for each query shape in the dataset (the score does not include the query itself). Figure 7(c): Average Precision versus Recall.



(a) **MPEG7 dataset**. Pairs of shapes from different classes. The complete dataset is composed of 8 classes of 5 elements.



(b) Recall vs Image Rank.



(c) Average Precision-Recall.

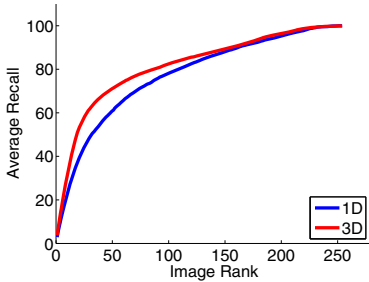
**Fig. 8. Retrieval results on MPEG7 database.** Figure 8(a): database overview. Figure 8(b): Average recall rate. Figure 8(c): Average Precision versus Recall.

### 5.2 3-D Shape Retrieval

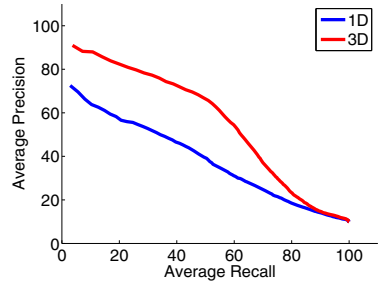
For surface, the domain  $\mathcal{E} = \Omega$  is the whole manifold. Hence, the first order quantiles  $Q_x(0)$  are discarded since they are zero, so that we handle  $d = 3$  -



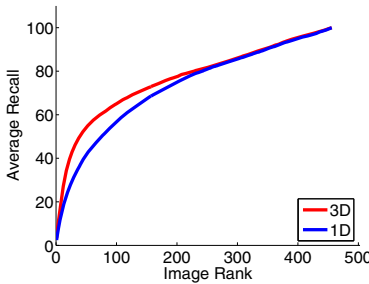
(a) McGill dataset of articulated and non-articulated objects [35] (respectively composed of 9 classes of 202 elements and 10 classes of 255 elements).



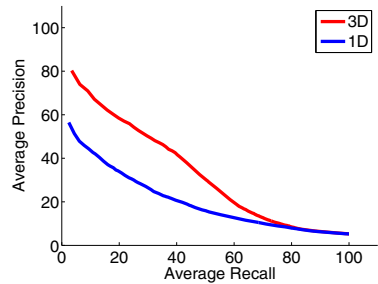
(b) Recall vs Image Rank on articulated dataset.



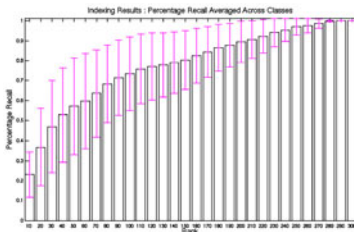
(c) Average Precision-Recall on articulated dataset.



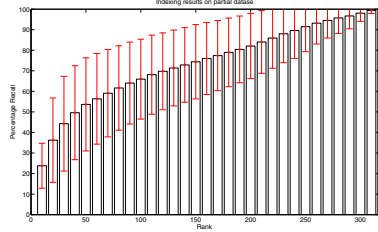
(d) Recall vs Image Rank on complete dataset.



(e) Average Precision-Recall on complete dataset.



(f) Average Recall on partial McGill database (14 classes) with [36] method.



(g) Average Recall with our approach.

**Fig. 9. Retrieval results on McGill database [35].** Figure 9(a): overview of the database. Figure 9(b) and 9(d): Average recall rate on articulated and complete dataset. Figure 9(c) and 9(e): Average Precision versus Recall on articulated and complete dataset. Figures 9(f) (from [36]) and 9(g): comparison of our approach with state-of-the-art method described in [36] on partial McGill Dataset (14 classes out of 19). Results are shown as average recall rate curve (plotted in black) along with the average intra-class standard deviation (in red).

dimensional geodesic statistics of  $n = 500$  points, considering from now *max* and *median* in addition with *mean* values of geodesic distances.

To evaluate the robustness of such global descriptor for 3-D shapes, we used the McGill dataset of 3-D articulated objects [35] (see Fig. 9(a) for an overview). Retrieval results are shown in Figs. 9(b) and 9(c). Again, it is clear that the combination of several geodesic distance characteristics achieves better retrieval results than considering only one. A comparison with state-of-the-art approach described in [36] is also provided in Figure 9, where we obtain similar results. Following the same protocol as in [36], retrieval results are given for a subset of the complete McGill Database (14 classes out of 19).

## 6 Conclusion

The first contribution of this paper is a generic framework to represent manifolds with statistical signatures based on geodesic distances, which are robust to bendings. The second contribution of this paper is an algorithm to compute a similarity measure between multi-dimensional joint distributions, which yields a fast approximation of the Wasserstein metric. This algorithm is applied to perform shape retrieval using our geodesic framework.

Our framework extends naturally to include additional information such as texture. One can indeed use a non-constant Riemannian metric that takes into account this information.

## References

1. Veltkamp, R.C., Latecki, L.: Properties and performance of shape similarity measures. In: Proc. of Conference on Data Science and Classification (2006)
2. Zhang, D.S., Lu, G.J.: Review of shape representation and description techniques. Pattern Recognition 37, 1–19 (2004)
3. Bustos, B., Keim, D.A., Saupe, D., Schreck, T., Vranić, D.V.: Feature-based similarity search in 3D object databases. ACM Comput. Surv. 37, 345–387 (2005)
4. Tangelder, J.W.H., Veltkamp, R.C.: A survey of content based 3D shape retrieval methods. Multimedia Tools Appl. 39, 441–471 (2008)
5. Teague, M.: Image analysis via the general theory of moments. Journal of the Optical Society of America 70, 920–930 (1980)
6. Teh, C.H., Chin, R.T.: On image analysis by the methods of moments. IEEE Trans. Patt. Anal. and Mach. Intell. 10, 496–513 (1988)
7. Liao, S., Pawlak, M.: On image analysis by moments. IEEE Trans. Patt. Anal. and Mach. Intell. 18, 254–266 (1996)
8. Zahn, C.T., Roskies, R.Z.: Fourier descriptors for plane closed curves. IEEE Transactions on Computer 21, 269–281 (1972)
9. Prokop, R.J., Reeves, A.P.: A survey of moment-based techniques for unoccluded object representation and recognition. CVGIP 54, 438–460 (1992)
10. Reuter, M., Wolter, F.E., Peinecke, N.: Laplace-spectra as fingerprints for shape matching. In: Symposium on Solid and Physical Modeling, pp. 101–106 (2005)
11. Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Shape distributions. ACM Transactions on Graphics 21, 807–832 (2002)
12. Ben Hamza, A., Krim, H.: Geodesic matching of triangulated surfaces. IEEE Trans. Image Proc. 15, 2249–2258 (2006)

13. Ion, A., Peyré, G., Haxhimusa, Y., Peltier, S., Kropatsch, W., Cohen, L.: Shape matching using the geodesic eccentricity transform - a study. In: Proc. Workshop of the Austrian Association for Pattern Recognition, OCG, pp. 97–104 (2007)
14. Ion, A., Artner, N., Peyré, G., López Mármol, S., Kropatsch, W., Cohen, L.: 3D shape matching by geodesic eccentricity. In: Proc. Workshop on Search in 3D. IEEE, Los Alamitos (2008)
15. Gorelick, L., Galun, M., Sharon, E., Basri, R., Brandt, A.: Shape representation and classification using the Poisson equation. *IEEE TPAMI* 28, 1991–2005 (2006)
16. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. *IEEE Trans. Patt. Anal. and Mach. Intell.* 24, 509–522 (2002)
17. Ling, H., Jacobs, D.W.: Shape classification using the inner-distance. *IEEE Trans. Patt. Anal. and Mach. Intell.* 29, 286–299 (2007)
18. Johnson, A., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. Patt. Anal. and Mach. Intell.* 21, 433–449 (1999)
19. Villani, C.: *Topics in Optimal Transportation*. American Mathematical Society, Providence (2003)
20. Rubner, Y., Tomasi, C., Guibas, L.J.: The Earth Mover’s Distance as a Metric for Image Retrieval. *International Journal of Computer Vision* 40, 99–121 (2000)
21. Grauman, K., Darrell, T.: Fast contour matching using approximate earth mover’s distance. In: Proc. of IEEE CVPR 2004, pp. I: 220–227 (2004)
22. Ling, H., Okada, K.: An Efficient Earth Mover’s Distance Algorithm for Robust Histogram Comparison. *IEEE Trans. PAMI* 29, 840–853 (2007)
23. Sethian, J.A.: A fast marching level set method for monotonically advancing fronts. *Proc. of the National Academy of Sciences* 93, 1591–1595 (1996)
24. Tsitsiklis, J.: Efficient Algorithms for Globally Optimal Trajectories. *IEEE Trans. on Automatic Control* 40, 1528–1538 (1995)
25. Kimmel, R., Sethian, J.: Computing Geodesic Paths on Manifolds. *Proc. of the National Academy of Sciences* 95, 8431–8435 (1998)
26. Bornemann, F., Rasch, C.: Finite-element discretization of static Hamilton-Jacobi equations based on a local variational principle. *Comput. Visual Sci.* 9 (2006)
27. Gonzalez, T.F.: Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science* 38, 293–306 (1985)
28. Peyré, G., Cohen, L.D.: Geodesic remeshing using front propagation. *International Journal of Computer Vision* 69, 145–156 (2006)
29. Veltkamp, R.C., Hagedoorn, M.: State of the art in shape matching, pp. 87–119 (2001)
30. Burkard, R., Dell’Amico, M., Martello, S.: *Assignment Problems*. SIAM, Philadelphia (2009)
31. Rabin, J., Peyré, G., Delon, J., Bernot, M.: Wasserstein Barycenter and its Application to Texture Mixing, <http://hal.archives-ouvertes.fr/hal-00476064/en/>
32. Bottou, L.: Online algorithms and stochastic approximations. In: Saad, D. (ed.) *Online Learning and Neural Networks*. Cambridge University Press, Cambridge (1998)
33. Pitié, F., Kokaram, A., Dahyot, R.: Automated colour grading using colour distribution transfer. In: *Computer Vision and Image Understanding* (2007)
34. Shirdhonkar, S., Jacobs, D.: Approximate Earth Mover’s Distance in linear time. In: Proc. CVPR 2008, pp. 1–8 (2008)
35. McGill 3-D shapes dataset, <http://www.cim.mcgill.ca/~shape/benchMark/>
36. Siddiqi, K., Zhang, J., Macrini, D., Shokoufandeh, A., Bouix, S., Dickinson, S.: Retrieving articulated 3-D models using medial surfaces. *Mach. Vision Appl.* 19, 261–275 (2008)

# Image Segmentation with Topic Random Field

Bin Zhao<sup>1</sup>, Li Fei-Fei<sup>2</sup>, and Eric P. Xing<sup>1</sup>

<sup>1</sup>School of Computer Science, Carnegie Mellon University

<sup>2</sup>Computer Science Department, Stanford University

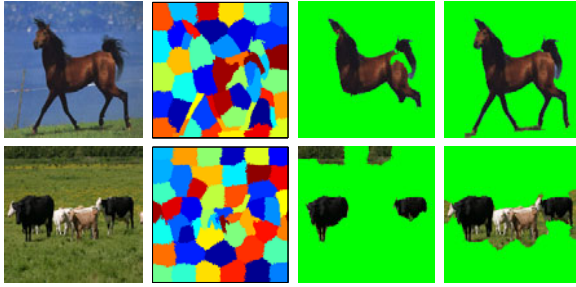
**Abstract.** Recently, there has been increasing interests in applying aspect models (e.g., PLSA and LDA) in image segmentation. However, these models ignore spatial relationships among local topic labels in an image and suffers from information loss by representing image feature using the index of its closest match in the codebook. In this paper, we propose Topic Random Field (TRF) to tackle these two problems. Specifically, TRF defines a Markov Random Field over hidden labels of an image, to enforce the spatial coherence between topic labels for neighboring regions. Moreover, TRF utilizes a noise channel to model the generation of local image features, and avoids the off-line process of building visual codebook. We provide details of variational inference and parameter learning for TRF. Experimental evaluations on three image data sets show that TRF achieves better segmentation performance.

## 1 Introduction

Image segmentation represents a fundamental problem in computer vision, which aims to cluster pixels in an image into distinct, semantically coherent and salient regions [1,2,3]. Solutions to image segmentation serves as the basis for a wide range of applications including object recognition, content-based image retrieval, video surveillance and object tracking [4].

Although geometry-based methods such as normalized cuts [1] remain an effective approach to image segmentation, motivated by the success of probabilistic aspect models, such as the probabilistic latent semantic analysis (PLSA) [5] and the latent Dirichlet allocation (LDA) [6], in text analysis and information retrieval, there has been a growing interest in applying such models for semantically-driven segmentation of natural images [7,8,9,10,11,12]. Among various advantages offered by these approaches, is their affordances for unsupervised training of representations of the latent aspects underlying a content-rich corpora, often known as *topics*, which can help define a semantically meaningful “content space” in which an image can lie. Thus the segmental results derived from an aspect model (also known as topic model) can be more reliant on content coherence, rather than mere spatial contiguity as in the spectrum methods. Other advantages include flexibility in capturing content granularity [7], and computational efficiency based on efficient approximate inference.

To apply those aspect models originally proposed for text data, it is necessary to first build a connection between an image and a text document. While text



**Fig. 1.** (Best viewed in color) Comparison of the spatial latent Dirichlet allocation (spatial LDA) model [13], one of the state-of-the-art aspect model for image analysis, and our proposed model, topic random field (TRF). First column shows the input images. Second column shows the input regions to spatial LDA and TRF provided by an over-segmentation method (normalized cut in this paper). The last two columns show the segmentation results of spatial LDA and TRF respectively. The first row indicates that by defining MRF over latent topics, TRF enforces spatial coherency over adjacent regions, while spatial LDA separates adjacent and semantically similar regions into two segments. The second row shows that using noise channel instead of codebook enables TRF to group two visually non-identical objects (black cow and white cow) from the same semantic class into the same category, while the spatial LDA model categorizes these two objects into different semantic classes.

documents are naturally composed from a vocabulary of distinctive words, an image is made from a collection of pixels, and there is no such obvious word-level representation for images. Conventionally, researchers extract various local features, for example, interest points detected by scale invariant saliency detector [14] as used in [13], and transform these local features to “visual words”, which play the same role as textual words in text analysis. Typically, after extracting local features from training images, a clustering is performed on the entire set of local features. Then a dictionary is constructed, with “words” being the centroids of the feature clusters. Based on this dictionary, each feature extracted from the image is then represented by the index of the most similar item (i.e. a visual word defined by the feature centroid) in the dictionary. Finally, analogous to text data, an image is represented as a collection of visual words, obtained by assigning every local feature an index in the visual dictionary.

Despite the success of modern low-level visual feature detectors, the aspect model built upon those local features suffers from several weaknesses. First, most existing aspect models regard an image as a bag of visual words, ignoring the spatial relationship between them. Although the spatial relationship between words in text documents might not severely affect content distillation, the spatial relationship between visual words are crucial for image understanding. For example, a scrambled collection of patches from a building image does not necessarily evoke the recognition of a building [11]. Most current work on aspect model of images ignores this important issue, hence might have compromised the final accuracy of the segmentation and recognition tasks. This contrasts the spectrum methods for which spatial contiguity is crucial in defining segmental

patterns. Second, representing each local image feature by the index of the item that is closest to it in the dictionary can result in severe loss of information. Due to the usual high dimensionality of local features extracted from images, it is impractical to build a large size dictionary that could enumerate all possible local features. Therefore, it is highly possible that even the closest matching visual word in the dictionary for a particular local feature instance can be quite different from the feature instance itself, and the matched visual word might even represent a mismatching content, thereby causing ambiguity in feature-instance versus visual-word matching. This phenomena has never been an issue in text modeling, where a word-instantiation in a document can be always unambiguously mapped to a word in the dictionary. We suspect that these two problems could seriously hinder the application of aspect models on image data.

In this paper, we propose a *Topic Random Field* (TRF) model for image segmentation, which improves over the basic LDA-style models, by defining a Markov Random Field (MRF) over hidden topic assignment of super-pixels in an image to enforce the spatial coherence between neighboring regions; and by employing a noise channel between visual words in the dictionary and instantiated super-pixels in the real image to better model the variance of local features. Specifically, instead of assuming that the latent topic assignments of every super-pixels in an image are generated independently according to a multinomial distribution, a TRF defines an MRF over the hidden super-pixels' labels to model their spatial relationship. Moreover, different from previous attempts, which first build a codebook off-line and then generate each local feature instantiation according to a multinomial distribution over word-index, a TRF generates each local feature instantiation as a corrupted or transformed version of a matching visual word in the codebook according to a noise-channel model, which allows explicit modeling and inference of the ambiguity of the matching between feature instantiation and feature prototypes (i.e., visual word). As a result, TRF avoids the problem of information loss during topic learning without building a large size codebook, and is significantly more robust to variability in the instantiations of local features corresponding to the same objects or common visual words due to variations in lighting, transformation, viewing angle, etc.

It should be noted that there has been some attempt in utilizing spatial relationships between topic labels to improve the performance of aspect models on image segmentation [13,11,15,16,17]. Probably the most related work to this paper is the spatial-LDA model [13], which also considers utilizing spatial consistency, by defining latent topic variables on over-segmented regions and enforcing all local patches within the region to share the same latent topic. In fact, we adopt a similar way of defining latent topic variables on over-segmented regions to enforce spatial consistency between local patches within the same region. However, in spatial-LDA model, the authors only consider the spatial consistency between adjacent local patches, while topic labels for over-segmented regions are assumed to be generated independently. Empirical comparison between spatial-LDA and TRF demonstrate the necessity of enforcing spatial consistency between adjacent over-segmented regions. Besides, in [11], the authors demonstrated that the



performance of PLSA can be improved by introducing an image-specific MRF to enforce the spatial coherence on the labels of the fine-grained local patches in an image. However, in that model the number of parameters grows linearly with the number of training images and the model is trained fully supervised. On the other hand, the number of parameters in TRF does not grow with the size of the training data because we apply a universally-parameterized MRF within the TRF over all images, which can be trained via a maximum likelihood principle in a fully unsupervised fashion. Applying a universal MRF rather than an image-specific one as in [11] is crucial to avoid overfitting and enable scalability. Moreover, [11] builds an MRF on local patches. Since there might be several hundreds of patches in one image, the resulting MRF is quite large; whereas our approach defines an MRF on over-segmented regions usually with homogeneous object-level contents, whose number in an image is around 50, and enforces the consistency among local semantically similar and adjacent patches by enforcing them to share the same latent topic. Therefore, the MRF in our model is much smaller than the one in [11], yet enforces the same amount of spatial consistency. In our empirical studies, we found that training TRF takes much less time than training the model in [11] on the same data set, which makes TRF more practical for web-scale image analysis.

Unlike the attempt on utilizing spatial relationships between topic labels to improve aspect model, as far as we are concerned, the noise channel presented in this paper is the first attempt in modeling visual feature generation without building a codebook in this topic-model based image analysis. Despite the fact that noise model could tolerate variability in the instantiations of local features due to variations in lighting, transformation, viewing angle, etc, using a noise channel also avoids the hassle of building a codebook off-line.

In summary, the main contributions of this paper can be highlighted as the follows: (1) The *Topic Random Field* provides a probabilistically sound framework for modeling spatial coherency within an aspect model. (2) TRF offers a more principled approach for addressing the ambiguity in feature-instance versus visual-word matching, and for codebook construction via unsupervised maximum likelihood learning during training the TRF (rather than via an off-line preprocessing). (3) The conjoint effect of a spatial MRF on topic labels and a noise-channel codebook lead to a segmental algorithm that takes into consideration of both semantic and spatial coherence, without any supervision. Figure 1 illustrates topic random field’s novelty by comparing the segmentation results of spatial-LDA and TRF.

The rest of this paper is organized as follows. We briefly review the image representation employed in aspect models for image segmentation problems, and describe the visual features we utilize in this paper. We introduce the topic random field model in Section 3. Section 4 presents the details of variational inference and parameter learning for this model. We give experimental results on three image data sets in Section 5, followed by conclusions in Section 6.

## 2 Preliminary: Image Representation

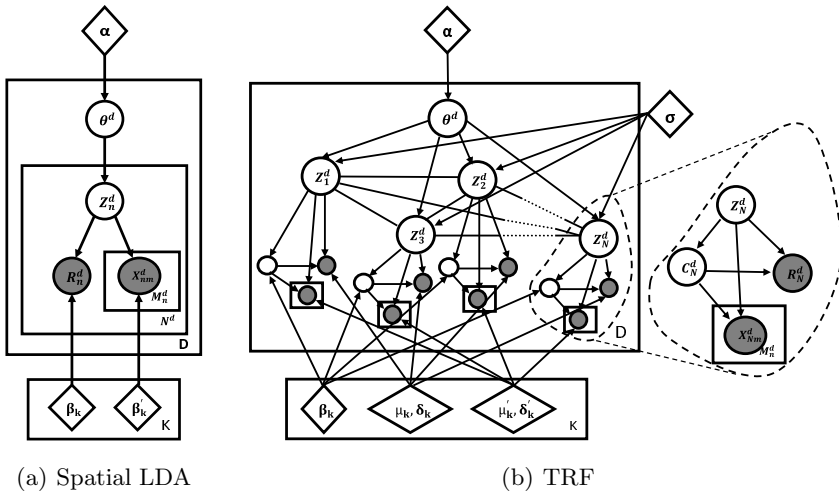
Given an image, TRF starts with an initial over-segmentation of the image by partitioning it into multiple homogeneous regions. To ensure that pixels in a region

belongs to the same object and avoid obtaining regions larger than the objects we want to segment, we start with an over-segmentation of the images using spectral clustering [1]. For each over-segmented region, we extract 4 types of region-level features: shape, color, location and texture. Specifically, the shape features include the centered object mask in a canonical  $32 \times 32$  frame, the size of the region, and the size of region's bounding box, which results in a 1027 dimensional vector [18]. The color features include the mean RGB value, its standard deviation and a color histogram. The location information extracted from each region is represented by a coarse  $8 \times 8$  absolute segmentation mask as well as the height of the top-most and bottom-most pixel in the region [18]. Finally, the texture features are average responses of filter banks in each region. Besides region-level features, we also extract pixel-level features within each segmented region. Specifically, we find a number of scale invariant interest points and describe them by SIFT [19].

### 3 Topic Random Field

In this section, we will introduce the *Topic Random Field* and explain in detail the generative process of this model. As discussed in the first section, TRF improves the spatial LDA model [13], a specially designed topic model for image segmentation, in two perspectives: the incorporation of an MRF over the hidden labels in the image and the introduction of a noise model for generating image features. To better understand the motivation of TRF, we first briefly describe the spatial LDA model as depicted in figure 2(a).

Given an image  $I^d$  ( $d \in \{1, 2, \dots, D\}$ ) and its over-segmented regions  $n = 1, 2, \dots, N^d$ , the spatial LDA model defines a latent topic  $z_n^d$  to represent the label of region  $n$ . Topics in image data have similar meanings as they do in text data: a



**Fig. 2.** Graphical model representation of Topic Random Field and comparison with spatial LDA model

topic represents category identity of an object, e.g., buildings, horses, cars, trees, etc. Suppose there are totally  $K$  topics within the image collection, then for each region  $n$ ,  $z_n^d \in \{1, \dots, K\}$ . Each topic  $z_n^d$  then generates a region-level feature  $R_n^d$ , for example the average filter responses in the region, and  $M_n^d$  pixel-level features  $\{\mathbf{X}_{nm}^d\}_{m=1}^{M_n^d}$ , such as the detected salient points described by SIFT. In order to do image segmentation using topic model, we first need to infer the hidden topic label for each over-segmented region and then group all the regions with the same topic label to form an object.

We will take an example to explain the generative process of spatial LDA: suppose we want to generate a “building” image  $I^d$ . First, we will draw a probability vector  $\boldsymbol{\theta}^d$  which determines what intermediate topics to select to generate each region of the image. For a building image,  $\boldsymbol{\theta}^d$  should privilege topics like “glasses”, “walls”, etc. Then, to create each region in the image, we determine a group of particular topics  $\{z_n^d\}_{n=1}^{N^d}$  out of the mixture of possible topics. For example, if a “glass” topic is selected, this will in turn give preference on some codewords that occur more frequently in glasses. Finally, we draw codewords  $R_n^d$  and  $\{\mathbf{X}_{nm}^d\}_{m=1}^{M_n^d}$  to describe the appearance of region  $n$ . The process of drawing both the topic and codewords will be repeated  $N^d$  times, eventually forming an entire bag of visual words that would construct an image of buildings.

### 3.1 Spatial MRF over Topic Assignments

The basic model ignores the spatial structure of the image, modeling its regions as independent draws from the topic mixing vector  $\boldsymbol{\theta}^d$ . However, the labels for adjacent regions tend to be strongly correlated in real images. TRF extends spatial LDA by enforcing spatial coherence among neighboring regions. Specifically, to enforce spatial coherence over hidden topic labels in our image model, we move from a multinomial distribution over hidden topics to a Markov Random Field. The topic random field, depicted as a generative model in Figure 2(b), introduces explicit couplings between the labels of adjacent regions in an image. This allows the TRF model the ability to capture local correlations that would be missed under the conditional independence assumption of spatial LDA. The transition from spatial LDA to TRF is equivalent to placing an MRF prior on hidden topic labels  $\mathbf{z}^d$ :

$$p(\mathbf{z}^d | \boldsymbol{\theta}^d, \sigma) = \frac{1}{A(\boldsymbol{\theta}^d, \sigma)} \exp \left[ \sum_n \sum_k z_{nk}^d \log \theta_k^d + \sum_{n \sim m} \sigma I(z_n^d = z_m^d) \right] \quad (1)$$

where  $I$  is the indicator function,  $n$  runs through all over-segmented regions in the image,  $k$  runs through all possible topics,  $n \sim m$  means that  $z_n^d$  and  $z_m^d$  are connected by an edge in the graphical model, and  $A(\boldsymbol{\theta}^d, \sigma)$  is the normalizing factor

$$A(\boldsymbol{\theta}^d, \sigma) = \sum_{\mathbf{z}^d} \exp \left[ \sum_n \sum_k z_{nk}^d \log \theta_k^d + \sum_{n \sim m} \sigma I(z_n^d = z_m^d) \right] \quad (2)$$

A positive value of  $\sigma$  awards configurations in which neighboring regions have the same label. Moreover, if we set  $\sigma = 0$ , i.e., assume the hidden topic labels are generated independently,  $A(\boldsymbol{\theta}^d, \sigma) = 1$  and  $\mathbf{z}^d$  follows a multinomial distribution parameterized by  $\boldsymbol{\theta}^d$ , and this gives us exactly the spatial LDA model.

Throughout this paper, we assume the Markov Random Field structure is known. Although structure learning over latent variables could be an interesting problem, it is not our intention to tackle this problem in current paper. The Markov Random Field is built by connecting a region with its nearest  $k$  neighbors.

### 3.2 Noise Channel over Codebook

Despite of the empirical success of aspect models on image data [12], one should be careful with the distinction between text data and image data. Representing each word by its index in the dictionary incurs no loss of information, and we could still recover that exact word using the index and the dictionary. However, due to the fact that there is no natural counterpart of words and dictionary in image data, we have to manually build a dictionary. Different from text data, representing each visual feature by the index of its most similar visual word in the dictionary will lose information about that particular local feature, since it is highly possible that there might not be an exact match in the dictionary we built. One would probably argue that we could alleviate this problem by building a large dictionary, to make sure every possible local detector has exact or close enough match in the dictionary. However, different from text word, visual words are usually high dimensional, to ensure each visual word has exact match in the dictionary would render the dictionary so large that no practical inference algorithm could solve the resulting model.

Therefore, the size of the codebook becomes crucial: a small codebook would incur heavy information loss, while a large codebook could render the model too difficult to solve. However, a closer look into the problem reveals that although it is not possible to exactly match every visual feature to a visual word in the dictionary, we could always find an entry in the dictionary such that the visual feature could be represented by this entry plus some noise. For example, given features extracted from a tree image, it is highly possible that we could extract similar features from another tree image. This intuition tells us that we could find several “prototype” visual features for an object, and model features extracted from the same object by these prototype features plus some noise. Therefore, each object is represented by a group of prototype features, and feature extracted from each individual image is the combination of prototype feature and noise.

To ease the description of the model, in the rest of this paper, we use  $\mathbf{x}_n^d$  to represent both region-level feature  $R_n^d$  and pixel-level features  $\{\mathbf{X}_{nm}^d\}_{m=1}^{M_n^d}$ . The generative process for visual features could then be modeled as a two-step process: first draw the prototype indicator  $c_n^d$  according to a multinomial distribution  $p(c_n^d | z_n^d, \beta)$ , then draw the visual feature  $\mathbf{x}_n^d$  using a noise model  $p(\mathbf{x}_n^d | c_n^d, z_n^d, \boldsymbol{\mu}, \delta)$ , where  $\boldsymbol{\mu}$  and  $\delta$  are parameters. Specifically, in this paper, we employ a Gaussian noise model, where  $\boldsymbol{\mu}$  is the mean vector and  $\delta^2$  is the variance. Suppose the number of possible prototype features for each object is  $L_k$ , then  $c_n^d \in \{1, \dots, L_k\}$ . For simplicity, we assume the number of different prototypes for all objects are the same, say  $L$ . Then the Gaussian noise model is

$$p(\mathbf{x}_n^d | c_n^d = l, z_n^d = k, \boldsymbol{\mu}, \delta) \propto \exp \left\{ -\frac{(\mathbf{x}_n^d - \boldsymbol{\mu}_{kl})^T (\mathbf{x}_n^d - \boldsymbol{\mu}_{kl})}{2\delta_{kl}^2} \right\} \quad (3)$$

Note that by introducing the noise model, we no longer need to build a codebook off-line. The prototype features are learned during the training process, and are stored in the mean vectors  $\boldsymbol{\mu}$ . This could also be understood as building a “codebook” online, where  $L$  is the size of the codebook for each object. The optimal value of  $L$  could be determined using Bayesian information criterion [20].

### 3.3 The Proposed Model

The generative process of *Topic Random Field* is as follows:

- For each image  $I^d$ , draw the prior distribution of  $\boldsymbol{\theta}^d$  according to a Dirichlet distribution parameterized by  $\boldsymbol{\alpha}$ ;
- Draw hidden topic labels  $\{z_1^d, \dots, z_{N^d}^d\}$  according to Markov random field parameterized by  $\boldsymbol{\theta}^d$ ;
- For each over-segmented region  $n \in \{1, \dots, N^d\}$ :
  - Draw a prototype appearance indicator  $c_n^d | z_n^d \sim \text{Mult}(\boldsymbol{\beta})$ ;
  - Draw region-level and pixel-level appearance features according to the noise model  $p(\mathbf{x}_n^d | c_n^d, z_n^d, \boldsymbol{\mu}, \delta)$

Putting the generative process together, the joint distribution of  $\{\boldsymbol{\theta}^d, \mathbf{z}^d, \mathbf{c}^d, \mathbf{x}^d\}$  given an image  $I^d$  can be written as

$$\begin{aligned}
 & p(\boldsymbol{\theta}^d, \mathbf{z}^d, \mathbf{c}^d, \mathbf{x}^d | \boldsymbol{\alpha}, \sigma, \boldsymbol{\beta}, \boldsymbol{\mu}, \delta) \\
 &= p(\boldsymbol{\theta}^d | \boldsymbol{\alpha}) p(\mathbf{z}^d | \boldsymbol{\theta}^d) \prod_{n=1}^{N^d} p(\mathbf{c}_n^d | \mathbf{z}_n^d, \boldsymbol{\beta}) p(\mathbf{x}_n^d | \mathbf{z}_n^d, \mathbf{c}_n^d, \boldsymbol{\mu}, \delta) \\
 &= \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K (\theta_k^d)^{\alpha_k - 1} \frac{1}{A(\boldsymbol{\theta}^d, \sigma)} \left( \prod_{n=1}^{N^d} \prod_{k=1}^K (\theta_k^d)^{z_{n,k}^d} \right) \exp \left[ \sum_{n \sim m} \sigma (\mathbf{z}_n^d)^T \mathbf{z}_m^d \right] \\
 & \cdot \prod_{n=1}^{N^d} \left\{ \prod_{k=1}^K \prod_{l=1}^L [\beta_{kl} p(\mathbf{x}_n^d | \boldsymbol{\mu}_{kl}, \delta_{kl})]^{z_{n,k}^d c_{n,l}^d} \right\}
 \end{aligned} \tag{4}$$

where we abuse the notation by defining  $z_{n,k}^d = 1$  if and only if  $z_n^d = k$ , and  $c_{n,l}^d = 1$  if and only if  $c_n^d = l$ .  $p(\mathbf{x}_n^d | \boldsymbol{\mu}_{kl}, \delta_{kl})$  is the noise model parameterized with  $(\boldsymbol{\mu}_{kl}, \delta_{kl})$ . After training the model, we label the region  $r$  with  $(z_r^d)^*$  such that

$$(z_r^d)^* = \arg \max_{z_r^d} p(\mathbf{x}_r^d | z_r^d) \tag{5}$$

The regions with the specific  $(z_r^d)^*$  constitute the interested object.

## 4 Variational Inference and Parameter Learning

The central challenge in using TRF is computing the posterior distribution of hidden variables given an image:  $p(\boldsymbol{\theta}^d, \mathbf{c}^d, \mathbf{z}^d | \mathbf{x}^d)$ . In general, this distribution is intractable to compute due to the dependence between  $\boldsymbol{\theta}^d$ ,  $\mathbf{c}^d$  and  $\mathbf{z}^d$ , once conditioned on some observations. Various variational inference algorithms have been proposed in the machine learning literature to solve this problem. In this paper, we employ mean field variational inference to efficiently obtain an approximation to this distribution. Specifically, mean field variational inference algorithm forms

a factorized distribution of the latent variables, parameterized by free variables known as variational parameters [21].

$$q(\theta^d, \mathbf{z}^d, \mathbf{c}^d | \gamma^d, \rho^d, \xi^d) = q(\theta^d | \gamma^d) \prod_{n=1}^{N^d} q(\mathbf{z}_n^d | \rho_n^d) q(\mathbf{c}_n^d | \xi_n^d) \tag{6}$$

where the Dirichlet parameters  $\gamma^d$  and the multinomial parameters  $(\rho_1^d, \dots, \rho_N^d)$ ,  $(\xi_1^d, \dots, \xi_N^d)$  are variational variables. These parameters are fit by minimizing the Kullback-Leibler (KL) divergence between the approximated and true posterior [21]. We begin with bounding the log likelihood of an image  $I^d$  by Jensen’s inequality. Specifically, we use variational EM algorithm to do inference and parameter learning for the TRF model. As shown in Algorithm 1, the E-step optimizes the variational parameters  $\{\gamma^d, \xi^d, \rho^d\}$  as follows<sup>1</sup>

$$\gamma_k^d = \alpha_k^d + \sum_{n=1}^{N^d} \rho_{nk}^d, \lambda^d = e^{|E^d| \sigma} \tag{7}$$

$$\xi_{nl}^d \propto \prod_{k=1}^K \left\{ \beta_{kl} \left( \frac{1}{2\pi \delta_{kl}^2} \right)^{\frac{m}{2}} \exp \left[ -\frac{(\mathbf{x}_n^d - \boldsymbol{\mu}_{kl})^T (\mathbf{x}_n^d - \boldsymbol{\mu}_{kl})}{2\delta_{kl}^2} \right] \right\}^{\rho_{nk}^d} \tag{8}$$

$$\begin{aligned} \rho_{nk}^d &\propto \exp[\Psi(\gamma_k^d) - \Psi(\sum_{k=1}^K \gamma_k^d) + \sum_{m \in \mathcal{N}(\mathcal{I}^d)} \rho_{mk}^d] \\ &\cdot \prod_{l=1}^L \left\{ \beta_{kl} \left( \frac{1}{2\pi \delta_{kl}^2} \right)^{\frac{m}{2}} \exp \left[ -\frac{(\mathbf{x}_n^d - \boldsymbol{\mu}_{kl})^T (\mathbf{x}_n^d - \boldsymbol{\mu}_{kl})}{2\delta_{kl}^2} \right] \right\}^{\xi_{nl}^d} \end{aligned} \tag{9}$$

and the M-step optimizes model parameters  $\{\alpha, \sigma, \beta, \mu, \delta\}$

$$\beta_{kl} \propto \sum_{d=1}^D \sum_{n=1}^{N^d} \xi_{nl}^d \rho_{nk}^d, \boldsymbol{\mu}_{kl} = \frac{\sum_{d=1}^D \sum_{n=1}^{N^d} \xi_{nl}^d \rho_{nk}^d \mathbf{x}_n^d}{\sum_{d=1}^D \sum_{n=1}^{N^d} \xi_{nl}^d \rho_{nk}^d} \tag{10}$$

$$\delta_{kl}^2 = \frac{\sum_{d=1}^D \sum_{n=1}^{N^d} \xi_{nl}^d \rho_{nk}^d (\mathbf{x}_n^d - \boldsymbol{\mu}_{kl})^T (\mathbf{x}_n^d - \boldsymbol{\mu}_{kl})}{m \sum_{d=1}^D \sum_{n=1}^{N^d} \xi_{nl}^d \rho_{nk}^d} \tag{11}$$

$$\sigma = \frac{1}{|E|} \log \frac{\sum_{d=1}^D \sum_{k=1}^K \sum_{n \sim m} \rho_{nk}^d \rho_{mk}^d}{\sum_{d=1}^D \frac{1}{\lambda^d}} \tag{12}$$

## 5 Experiments

In this section, we show the empirical performance of topic random field for image segmentation, both qualitatively and quantitatively.

### 5.1 Data Sets

We use three data sets in our experiments, which are selected to cover a wide range of properties. Specifically, those data sets include

- **Weizmann data set** [22]. The data set contains 328 images of horses with different poses, sizes, face directions, backgrounds and illumination conditions. Each image has a ground truth segmentation that labels out the horse. There is only one horse in each image, and there is a single object category in the data set: horse.

---

<sup>1</sup> Here we omit the details due to the space limit. The derivation of variational inference and parameter learning for TRF is provided in the supplemental material.

---

**Algorithm 1.** Variational EM for topic random field

---

**repeat****E-step:** For each image  $I^d$ , update  $\{\gamma^d, \lambda^d, \xi^d, \rho^d\}$  using equations (7), (8), and (9);**M-step:** Update  $\{\sigma, \beta, \mu, \delta\}$  using equations (10), (11), (12), and update  $\alpha$  using the linear-time Newton-Raphson algorithm described in (6).**until** The increase of log likelihood between two consecutive iterations is less than  $\epsilon$ 

---

- **Microsoft object recognition data set** [23]. This data set involves 182 images of cows, facing three different directions: left, right and front. Moreover, some cow pictures also contain multiple instances and significant occlusions. Similar to the Weizmann data set, there is a single object category in the data set, but there might be multiple objects in one image.
- **MSRC pixel-wise labeled image database** [2]. There are 240,  $213 \times 320$  pixel images in this data set. Each pixel belongs to one of 13 semantic classes or to the void class. There are multiple objects in one image, and multiple object categories in the data set.

## 5.2 Experimental Setups and Comparisons

We have conducted comprehensive performance evaluations by testing our method under different circumstances. Specifically, to better understand the effect of introducing MRF on latent topics to enforce spatial consistency and use of noise model to better model image feature generation, we study the model adding only MRF on latent topics and adding only noise model separately, and compare with the TRF model. We use the spatial LDA model [13], which is state-of-the-art aspect model for image segmentation, as baseline and also compare with spectral clustering. The algorithms that we evaluated are listed below.

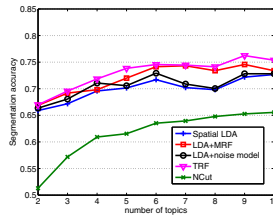
- **Spatial LDA** [13]. The implementation is the same as in [13]. We use the same region-level and pixel-level features as in our TRF model.
- **LDA+MRF**. This model is based on spatial LDA [13], with the only modification of introducing a Markov random field on the latent topics. Thus, this model could be viewed as the TRF model without noise channel. For each image  $I^d$ , we set  $L = 20$  and build a Markov random field on  $\mathbf{z}^d$  by connecting each  $\mathbf{z}_n^d$  with its nearest 4 neighbors.
- **LDA+noise** Similar with LDA+MRF, this model adds a noise channel in the spatial LDA model. Hence, this model could be regarded as the TRF model without Markov random field on the latent topics.
- **TRF**. We build MRF for each image in the same way as LDA+MRF.
- **Normalized cuts. (NCut)** [1]. The implementation code is downloaded from <http://www.cis.upenn.edu/~jshi/software/>.

---

<sup>2</sup> <http://research.microsoft.com/vision/cambridge/recognition>

### 5.3 Image Segmentation Results

Since the Weizman data set provides ground truth segmentations, we could assess the segmentation result quantitatively. Regions sharing the same latent topics  $z$  are grouped into the same segment, and the percentage of pixels in agreement with the ground truth segmentation is used to measure the performance of segmentation algorithms. We match the topic that resulted in highest segmentation accuracy as the object, and other topics as background. The segmentation accuracy results are shown in figure 3, from which we could see that both LDA+MRF and LDA+noise model result in higher accuracy than spatial LDA model, and topic random field produces the highest segmentation accuracy. Also, the comparison between LDA+MRF and LDA+noise model shows that the Markov random field defined over latent topic variables improves the accuracy more. It should be noted that our result is not directly comparable to that of the state-of-the-art image segmentation methods, as we did not engineer our image features much. The message here is that spatial consistency and a better model of image feature generation are crucial for the success of aspect models in image analysis.



**Fig. 3.** Segmentation accuracy of normalized cut, spatial LDA, LDA+MRF, LDA+noise, and TRF on the Weizmann horse data set



**Fig. 4.** (Best viewed in color). Segmentation results of horses. From left to right: original image, segmentation result of spatial LDA and TRF. The regions in white are the segmentations of the animals. The regions in black stand for background.





**Fig. 5.** (Best viewed in color). Segmentation results of the MSRC database. From left to right: original image, segmentation result of spatial LDA and TRF.



**Fig. 6.** (Best viewed in color). Segmentation results of cows. From left to right: original image, segmentation result of spatial LDA and TRF.

To better compare the performance of TRF with LDA, we show in figures 4,5,6 the segmentation results on the three data sets, where we have set the number of topics to 4, 12, 4 respectively. From these segmentation results, we could see that one major problem with spatial LDA is that it is more likely to separate parts from the same object into different segments. For example, in the Weizmann horse data, spatial LDA constantly separates the body and legs of a horse into different groups. However, the segmentation results of TRF does not show this phenomenon. Therefore, we argue that enforcing spatial coherence between adjacent regions via MRF avoids separating parts of the same object into different groups. Moreover, from the results on cows data set, we see that spatial LDA is more likely to segment cows with different colors or facing different directions into separate groups. However, by introducing a simple Gaussian noise model for generating image features, TRF is significantly more robust to variability in the instantiations of local features corresponding to the same objects due to variations in lighting, transformation, viewing angle, etc.

## 6 Conclusions

We propose *Topic Random Field* (TRF) for image segmentation. The TRF model improves over the LDA-style models by defining a Markov Random Field (MRF) over hidden topic assignment of super-pixels in an image to enforce the spatial coherence between neighboring regions, and by employing a noise channel between visual words in the dictionary and instantiated super-pixels in the real image to better model the variance of local features. Empirical studies on three image data sets demonstrate the improvement of our model in image segmentation over the LDA-style model.

**Acknowledgments.** EPX is supported by NSF IIS-0713379, DBI-0640543, DBI-0546594, Career Award, ONR N000140910758, DARPA NBCH1080007, Alfred P. Sloan Foundation. LFF is supported by NSF IIS-0845230, Career Award.

## References

1. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 888–905 (2000)
2. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, 603–619 (2002)
3. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. *International Journal of Computer Vision* 59, 167–181 (2004)
4. Forsyth, D., Ponce, J.: *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference (2002)
5. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn.* 42, 177–196 (2001)
6. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. *J. Mach. Learn. Res.* 3, 993–1022 (2003)
7. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 524–531 (2005)

8. Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A.: Learning object categories from google's image search. In: Proceedings of the Tenth IEEE International Conference on Computer Vision, pp. 1816–1823 (2005)
9. Russell, B., Freeman, W., Efros, A., Sivic, J., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1605–1614 (2006)
10. Sivic, J., Russell, B., Efros, A., Zisserman, A., Freeman, W.: Discovering object categories in image collections. In: Proceedings of the Tenth International Conference on Computer Vision (2005)
11. Verbeek, J., Triggs, B.: Region classification with markov field aspect models. In: Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
12. Li, L.J., Socher, R., Fei-Fei, L.: Towards total scene understanding: classification, annotation and segmentation in an automatic framework. In: Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2009)
13. Cao, L., Fei-Fei, L.: Spatially coherent latent topic model for concurrent object segmentation and classification. In: Proceedings of IEEE International Conference on Computer Vision (2007)
14. Kadir, T., Brady, M.: Saliency, scale and image description. *Int. J. Comput. Vision* 45, 83–105 (2001)
15. Sudderth, E., Jordan, M.: Shared segmentation of natural scenes using dependent pitman-yor processes. In: Proceedings of Neural Information Processing Systems (2008)
16. Andreetto, M., Zelnik-Manor, L., Perona, P.: Unsupervised learning of categorical segments in image collections. In: Proceedings of IEEE International Conference on Computer Vision (2008)
17. Levin, A., Weiss, Y.: Learning to combine bottom-up and top-down segmentation. *Int. J. Comput. Vision* 81, 105–118 (2009)
18. Malisiewicz, T., Efros, A.: Recognition by association via learning per-exemplar distances. In: Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2008)
19. Lowe, D.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60, 91–110 (2004)
20. Schwarz, G.: Estimating the dimension of a model. *The annals of statistics* 6, 461–464 (1978)
21. Wainwright, M., Jordan, M.: *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc. (2008)
22. Borenstein, E., Ullman, S.: Learning to segment. In: Pajdla, T., Matas, J(G.) (eds.) *ECCV 2004*. LNCS, vol. 3023, pp. 315–328. Springer, Heidelberg (2004)
23. Winn, J., Criminisi, A., Minka, T.: Object categorization by learned universal visual dictionary. In: Proceedings of IEEE International Conference on Computer Vision (2005)

# Author Index

- Abugharbieh, Rafeef IV-651  
Adler, Amir II-622  
Aeschliman, Chad II-594  
Agapito, Lourdes II-15, IV-283,  
IV-297  
Agarwal, Sameer II-29  
Agrawal, Amit I-100, II-237, III-129  
Ahuja, Narendra II-223, IV-87,  
VI-393, V-644  
Ai, Haizhou VI-238  
Alahari, Karteek IV-424  
Albarelli, Andrea V-519  
Alexe, Bogdan IV-452, V-380  
Aloimonos, Y. II-506  
Aloimonos, Yiannis V-127  
Alpert, Sharon IV-750  
Alterovitz, Ron III-101  
Andriyenko, Anton I-466  
Angst, Roland III-144  
Appia, Vikram I-73, VI-71  
Arbelaez, Pablo IV-694  
Arora, Chetan III-552  
Arras, Kai O. V-296  
Åström, Kalle II-114  
Avidan, Shai V-268  
Avraham, Tamar V-99  
Ayazoglu, Mustafa II-71
- Baatz, Georges VI-266  
Babenko, Boris IV-438  
Bae, Egil VI-379  
Bagdanov, Andrew D. VI-280  
Bagnell, J. Andrew VI-57  
Bai, Jiamin II-294  
Bai, Xiang III-328, V-15  
Bai, Xue V-617  
Bajcsy, Ruzena III-101  
Baker, Simon I-243  
Balikai, Anupriya IV-694  
Banerjee, Subhashis III-552  
Bao, Hujun V-422  
Baraniuk, Richard G. I-129  
Bar-Hillel, Aharon IV-127  
Barinova, Olga II-57
- Barnes, Connelly III-29  
Barreto, João P. IV-382  
Bartoli, Adrien II-15  
Basri, Ronen IV-750  
Baust, Maximilian III-580  
Behmo, Régis IV-171  
Belongie, Serge I-591, IV-438  
Ben, Shenglan IV-44  
BenAbdelkader, Chiraz VI-518  
Ben-Ezra, Moshe I-59  
Berg, Alexander C. I-663, V-71  
Berg, Tamara L. I-663  
Bernal, Hector I-762  
Bhakta, Vikrant VI-405  
Bischof, Horst III-776, VI-29, V-29  
Bitsakos, K. II-506  
Bizheva, Kostadinka K. III-44  
Black, Michael J. I-285  
Blanz, Volker I-299  
Boben, Marko V-687  
Boley, Daniel IV-722  
Boltz, Sylvain III-692  
Boucher, Jean-Marc IV-185, IV-764  
Boult, Terrance III-481  
Bourdev, Lubomir VI-168  
Bowden, Richard VI-154  
Boyer, Edmond IV-326  
Boykov, Yuri VI-379, V-211  
Bradski, Gary V-658  
Brandt, Jonathan VI-294  
Brandt, Sami S. IV-666  
Branson, Steve IV-438  
Bregler, Christoph VI-140  
Breitenreicher, Dirk II-494  
Brendel, William II-721  
Breuer, Pia I-299  
Bronstein, Alex III-398  
Bronstein, Alexander M. II-197  
Bronstein, Michael II-197, III-398  
Brown, Michael S. VI-323  
Brox, Thomas I-438, VI-168, V-282  
Bruhn, Andrés IV-568  
Bu, Jiajun V-631  
Burgoon, Judee K. VI-462

- Burschka, Darius II-183  
 Byröd, Martin II-114  
  
 Cagniard, Cedric IV-326  
 Cai, Qin III-229  
 Calonder, Michael IV-778  
 Camps, Octavia II-71  
 Cannons, Kevin J. IV-511  
 Cao, Yang V-729  
 Caplier, Alice I-313  
 Castellani, Umberto VI-15  
 Chandraker, Manmohan II-294  
 Chao, Hongyang III-342  
 Charpiat, Guillaume V-715  
 Chaudhry, Rizwan II-735  
 Chellappa, Rama I-129, III-286, V-547  
 Chen, Chih-Wei II-392  
 Chen, Chun V-631  
 Chen, David VI-266  
 Chen, Jiansheng IV-44  
 Chen, Jiun-Hung III-621  
 Chen, Siqi V-715  
 Chen, Weiping III-496  
 Chen, Xiaowu IV-101  
 Chen, Xilin I-327, II-308  
 Chen, Yu III-300  
 Chen, Yuanhao V-43  
 Cheong, Loong-Fah III-748  
 Chia, Liang-Tien I-706, IV-1  
 Chin, Tat-Jun V-533  
 Cho, Minsu V-492  
 Choi, Wongun IV-553  
 Christensen, Marc VI-405  
 Chua, Tat-Seng IV-30  
 Chum, Ondřej III-1  
 Chung, Albert C.S. III-720  
 Cipolla, Roberto III-300  
 Clausi, David A. III-44  
 Clipp, Brian IV-368  
 Cohen, Laurent D. V-771  
 Cohen, Michael I-171  
 Collins, Robert T. V-324  
 Collins, Roderic I-549, II-664  
 Courchay, Jérôme II-85  
 Cremers, Daniel III-538, V-225  
 Criminisi, Antonio III-510  
 Cristani, Marco II-378, VI-15  
 Cucchiara, Rita VI-196  
 Curless, Brian I-171, VI-364  
  
 Dai, Shengyang I-480  
 Dai, Yuchao IV-396  
 Dalalyan, Arnak II-85, IV-171  
 Dammertz, Holger V-464  
 Darrell, Trevor I-677, IV-213  
 Davies, Ian III-510  
 Davis, Larry S. II-693, IV-199, VI-476  
 Davison, Andrew J. III-73  
 De la Torre, Fernando II-364  
 Del Bue, Alessio III-87, IV-283, IV-297  
 Deng, Jia V-71  
 Deselaers, Thomas IV-452, V-380  
 Di, Huijun IV-525  
 Dickinson, Sven II-480, V-183, V-603  
 Dilsizian, Mark VI-462  
 Ding, Chris III-762, IV-793, VI-126  
 Ding, Lei IV-410  
 Ding, Yuanyuan I-15  
 Di Stefano, Luigi III-356  
 Dodgson, Neil A. III-510  
 Domokos, Csaba II-777  
 Dong, Zilong V-422  
 Donoser, Michael V-29  
 Douze, Matthijs I-522  
 Dragon, Ralf II-128  
 Duan, Genquan VI-238  
 Dunn, Enrique IV-368  
  
 Ebert, Sandra I-720  
 Efros, Alexei A. II-322, IV-482  
 Eichel, Justin A. III-44  
 Eichner, Marcin I-228  
 Elad, Michael II-622  
 Elmoataz, Abderrahim IV-638  
 Endres, Ian V-575  
 Eskin, Yulia V-183  
 Ess, Andreas I-397, I-452  
  
 Fablet, Ronan IV-185, IV-764  
 Fan, Jialue I-411, I-480  
 Fang, Tian II-1  
 Farenzena, Michela II-378  
 Farhadi, Ali IV-15  
 Fayad, João IV-297  
 Fazly, Afsaneh V-183  
 Fei-Fei, Li II-392, V-71, V-785  
 Fergus, Rob I-762, VI-140  
 Fermüller, C. II-506  
 Fernández, Carles II-678  
 Ferrari, Vittorio I-228, IV-452, V-380

- Fidler, Sanja V-687  
 Fieguth, Paul W. III-44  
 Finckh, Manuel V-464  
 Finkelstein, Adam III-29  
 Fite-Georgel, Pierre IV-368  
 Fitzgibbon, Andrew I-776  
 Fleet, David J. III-243  
 Flint, Alex V-394  
 Forsyth, David IV-15, IV-227,  
     VI-224, V-169  
 Fowlkes, Charless IV-241  
 Frahm, Jan-Michael II-142, IV-368  
 Franke, Uwe IV-582  
 Fraundorfer, Friedrich IV-269  
 Freeman, William T. III-706  
 Freifeld, Oren I-285  
 Fritz, Mario IV-213  
 Fua, Pascal III-58, III-370,  
     III-635, IV-778  
 Fuh, Chiou-Shann VI-84  
 Fusiello, Andrea I-790, V-589
- Gall, Juergen I-620, III-425  
 Gallagher, Andrew V-169  
 Gallup, David III-229, IV-368  
 Galun, Meirav IV-750  
 Gammeter, Stephan I-734  
 Gao, Shenghua IV-1  
 Gao, Wen I-327, II-308  
 Gao, Yongsheng III-496  
 Ge, Weina V-324  
 Gehler, Peter I-143, VI-98  
 Ghanem, Bernard II-223  
 Gherardi, Riccardo I-790  
 Glocker, Ben III-272  
 Godec, Martin III-776  
 Goldberg, Chen IV-127  
 Goldluecke, Bastian V-225  
 Goldman, Dan B. III-29  
 Gong, Leiguang IV-624  
 Gong, Yihong VI-434  
 González, Jordi II-678, VI-280  
 Gopalan, Raghuraman III-286  
 Gould, Stephen II-435, IV-497, V-338  
 Grabner, Helmut I-369  
 Gray, Douglas VI-434  
 Gryn, Jacob M. IV-511  
 Grzeszczuk, Radek VI-266  
 Gu, Chunhui V-408  
 Gu, Steve III-663
- Gu, Xianfeng V-672  
 Gualdi, Giovanni VI-196  
 Guan, Peng I-285  
 Guillaumin, Matthieu I-634  
 Guo, Huimin VI-476  
 Guo, Yanwen III-258  
 Gupta, Abhinav IV-199, IV-482  
 Gupta, Ankit I-171  
 Gupta, Mohit I-100
- Hager, Gregory D. II-183  
 Hall, Peter IV-694  
 Hamarneh, Ghassan IV-651  
 Han, Hu II-308  
 Han, Mei II-156  
 Han, Tony X. III-200, III-748  
 Harada, Tatsuya IV-736  
 Hartley, Richard III-524  
 Hauberg, Søren I-425, VI-43  
 Havlena, Michal II-100  
 He, Jinping IV-44  
 He, Kaiming I-1  
 He, Mingyi IV-396  
 He, Xuming IV-539  
 Hebert, Martial I-508, I-536,  
     IV-482, VI-57  
 Hedau, Varsha VI-224  
 Heibel, T. Hauke III-272  
 Heikkilä, Janne I-327, V-366  
 Hejrati, Mohsen IV-15  
 Hel-Or, Yacov II-622  
 Hesch, Joel A. IV-311  
 Hidane, Moncef IV-638  
 Hinton, Geoffrey E. VI-210  
 Hirzinger, Gerhard II-183  
 Hockenmaier, Julia IV-15  
 Hoiem, Derek VI-224, V-575  
 Hoogs, Anthony I-549, II-664  
 Horaud, Radu V-743  
 Horbert, Esther I-397  
 Hou, Tingbo III-384  
 Hsu, Gee-Sern I-271  
 Hu, Yiqun I-706  
 Hua, Gang I-243, III-200  
 Huang, Chang I-383, III-314  
 Huang, Dong II-364  
 Huang, Heng III-762, IV-793, VI-126  
 Huang, Junzhou III-607, IV-624  
 Huang, Thomas S. III-566, VI-490,  
     V-113, V-141

- Hung, Yi-Ping I-271  
 Huttenlocher, Daniel P. II-791  
  
 Idrees, Haroon III-186  
 Ik Cho, Nam II-421  
 Ikizler-Cinbis, Nazli I-494  
 Ilic, Slobodan IV-326  
 Ilstrup, David I-200  
 Ip, Horace H.S. VI-1  
 Isard, Michael I-648, III-677  
 Ishiguro, Hiroshi VI-337  
 Ito, Satoshi II-209, V-701  
 Ivanov, Yuri II-735  
  
 Jain, Arpit IV-199  
 Jamieson, Michael V-183  
 Jen, Yi-Hung IV-368  
 Jégou, Hervé I-522  
 Jeng, Ting-Yueh I-605  
 Ji, Qiang VI-532  
 Jia, Jiaya I-157, V-422  
 Jiang, Lin VI-504  
 Jiang, Xiaoyue IV-58  
 Jin, Xin IV-101  
 Johnson, Micah K. I-31  
 Johnson, Tim IV-368  
 Jojic, Nebojsa VI-15  
 Joshi, Neel I-171  
 Jung, Kyomin II-535  
 Jung, Miyoun I-185  
  
 Kak, Avinash C. II-594  
 Kalra, Prem III-552  
 Kankanhalli, Mohan IV-30  
 Kannala, Juho V-366  
 Kapoor, Ashish I-243  
 Kappes, Jörg Hendrik III-735  
 Kato, Zoltan II-777  
 Katti, Harish IV-30  
 Ke, Qifa I-648  
 Kembhavi, Aniruddha II-693  
 Kemelmacher-Shlizerman, Ira I-341  
 Keriven, Renaud II-85  
 Keutzer, Kurt I-438  
 Khuwuthyakorn, Pattaraporn II-636  
 Kim, Gunhee V-85  
 Kim, Hyeongwoo I-59  
 Kim, Jaewon I-86  
 Kim, Minyoung III-649  
 Kim, Seon Joo VI-323  
  
 Kim, Tae-Kyun III-300  
 Knopp, Jan I-748  
 Kohli, Pushmeet II-57, II-535,  
 III-272, V-239  
 Kohno, Tadayoshi VI-364  
 Kokkinos, Iasonas II-650  
 Kolev, Kalin III-538  
 Koller, Daphne II-435, IV-497, V-338  
 Kolmogorov, Vladimir II-465  
 Komodakis, Nikos II-520  
 Koo, Hyung Il II-421  
 Köser, Kevin VI-266  
 Krömer, Oliver II-566  
 Krupka, Eyal IV-127  
 Kubota, Susumu II-209, V-701  
 Kulikowski, Casimir IV-624  
 Kulis, Brian IV-213  
 Kuniyoshi, Yasuo IV-736  
 Kuo, Cheng-Hao I-383  
 Kwatra, Vivek II-156  
  
 Ladický, L'ubor IV-424, V-239  
 Lalonde, Jean-François II-322  
 Lampert, Christoph H. II-566, VI-98  
 Lanman, Douglas I-86  
 Lao, Shihong VI-238  
 Larlus, Diane I-720  
 Latecki, Longin Jan III-411, V-450,  
 V-757  
 Lauze, François VI-43  
 Law, Max W.K. III-720  
 Lawrence Zitnick, C. I-171  
 Lazarov, Maxim IV-72  
 Lazebnik, Svetlana IV-368, V-352  
 LeCun, Yann VI-140  
 Lee, David C. I-648  
 Lee, Jungmin V-492  
 Lee, Kyoung Mu V-492  
 Lee, Ping-Han I-271  
 Lee, Sang Wook IV-115  
 Lefort, Riwal IV-185  
 Leibe, Bastian I-397  
 Leistner, Christian III-776, VI-29  
 Lellmann, Jan II-494  
 Lempitsky, Victor II-57  
 Lensch, Hendrik P.A. V-464  
 Leonardis, Aleš V-687  
 Lepetit, Vincent III-58, IV-778  
 Levi, Dan IV-127  
 Levin, Anat I-214

- Levinshtein, Alex II-480  
 Lewandowski, Michal VI-547  
 Lézoray, Olivier IV-638  
 Li, Ang III-258  
 Li, Chuan IV-694  
 Li, Hanxi II-608  
 Li, Hongdong IV-396  
 Li, Kai V-71  
 Li, Na V-631  
 Li, Ruonan V-547  
 Li, Yi VI-504  
 Li, Yin III-790  
 Li, Yunpeng II-791  
 Li, Zhiwei IV-157  
 Lian, Wei V-506  
 Lian, Xiao-Chen IV-157  
 Lim, Yongsub II-535  
 Lin, Dahua I-243  
 Lin, Liang III-342  
 Lin, Yen-Yu VI-84  
 Lin, Zhe VI-294  
 Lin, Zhouchen I-115, VI-490  
 Lindenbaum, Michael V-99  
 Ling, Haibin III-411  
 Liu, Baiyang IV-624  
 Liu, Ce III-706  
 Liu, Jun VI-504  
 Liu, Risheng I-115  
 Liu, Shuaicheng VI-323  
 Liu, Siying II-280  
 Liu, Tyng-Luh VI-84  
 Liu, Wenyu III-328, V-15  
 Liu, Xiaoming I-354  
 Liu, Xinyang III-594  
 Liu, Xiuwen III-594  
 Liu, Yazhou I-327  
 Livne, Micha III-243  
 Lobaton, Edgar III-101  
 Lourakis, Manolis I.A. II-43  
 Lovegrove, Steven III-73  
 Lu, Bao-Liang IV-157  
 Lu, Zhiwu VI-1  
 Lucey, Simon III-467  
 Lui, Lok Ming V-672  
 Luo, Jiebo V-169  
 Luo, Ping III-342  
  
 Ma, Tianyang V-450  
 Maheshwari, S.N. III-552  
 Mair, Elmar II-183  
 Maire, Michael II-450  
 Maji, Subhransu VI-168  
 Majumder, Aditi IV-72  
 Makadia, Ameesh V-310  
 Makris, Dimitrios VI-547  
 Malik, Jitendra VI-168, V-282  
 Manduchi, Roberto I-200  
 Mansfield, Alex I-143  
 Marcombes, Paul IV-171  
 Mario Christoudias, C. I-677  
 Marks, Tim K. V-436  
 Matas, Jiří III-1  
 Matikainen, Pyry I-508  
 Matsushita, Yasuyuki II-280  
 Matthews, Iain III-158  
 McCloskey, Scott I-15, VI-309  
 Meer, Peter IV-624  
 Mehrani, Paria V-211  
 Mehran, Ramin III-439  
 Mei, Christopher V-394  
 Mensink, Thomas IV-143  
 Metaxas, Dimitris III-607, VI-462  
 Michael, Nicholas VI-462  
 Micheals, Ross III-481  
 Mikami, Dan III-215  
 Mikulík, Andrej III-1  
 Miller, Eric V-268  
 Mio, Washington III-594  
 Mirmehdi, Majid IV-680, V-478  
 Mitra, Niloy J. III-398  
 Mitzel, Dennis I-397  
 Mnih, Volodymyr VI-210  
 Monroy, Antonio V-197  
 Montoliu, Raúl IV-680  
 Moore, Brian E. III-439  
 Moorthy, Anush K. V-1  
 Morellas, Vassilios IV-722  
 Moreno-Noguer, Francesc III-58, III-370  
 Mori, Greg II-580, V-155  
 Morioka, Nobuyuki I-692  
 Moses, Yael III-15  
 Mourikis, Anastasios I. IV-311  
 Mu, Yadong III-748  
 Mukaigawa, Yasuhiro I-86  
 Müller, Thomas IV-582  
 Munoz, Daniel VI-57  
 Murino, Vittorio II-378, VI-15  
 Murray, David V-394



- Nadler, Boaz IV-750  
 Nagahara, Hajime VI-337  
 Nakayama, Hideki IV-736  
 Narasimhan, Srinivasa G. I-100, II-322  
 Nascimento, Jacinto C. III-172  
 Navab, Nassir III-272, III-580  
 Nayar, Shree K. VI-337  
 Nebel, Jean-Christophe VI-547  
 Neumann, Ulrich III-115  
 Nevatia, Ram I-383, III-314  
 Ng, Tian-Tsong II-280, II-294  
 Nguyen, Huu-Giao IV-764  
 Niebles, Juan Carlos II-392  
 Nielsen, Frank III-692  
 Nielsen, Mads IV-666, VI-43  
 Nishino, Ko II-763  
 Nowozin, Sebastian VI-98  
 Nunes, Urbano IV-382
- Obrador, Pere V-1  
 Oh, Sangmin I-549  
 Oliver, Nuria V-1  
 Ommer, Björn V-197  
 Orr, Douglas III-510  
 Ostermann, Joern II-128  
 Otsuka, Kazuhiro III-215  
 Oxholm, Geoffrey II-763  
 Özuysal, Mustafa III-58, III-635
- Packer, Ben V-338  
 Pajdla, Tomas I-748  
 Pajdla, Tomáš II-100  
 Paladini, Marco II-15, IV-283  
 Pantic, Maja II-350  
 Papamichalis, Panos VI-405  
 Papanikolopoulos, Nikolaos IV-722  
 Paris, Sylvain I-31  
 Park, Dennis IV-241  
 Park, Hyun Soo III-158  
 Park, Johnny II-594  
 Patel, Ankur VI-112  
 Patras, Ioannis II-350  
 Patterson, Donald IV-610  
 Pätz, Torben V-254  
 Pavlovic, Vladimir III-649  
 Payet, Nadia V-57  
 Pedersen, Kim Steenstrup I-425  
 Pedersoli, Marco VI-280  
 Pele, Ofir II-749  
 Pellegrini, Stefano I-452
- Perdoch, Michal III-1  
 Pérez, Patrick I-522  
 Perina, Alessandro VI-15  
 Perona, Pietro IV-438  
 Perronnin, Florent IV-143  
 Petersen, Kersten IV-666  
 Peyré, Gabriel V-771  
 Pfister, Hanspeter II-251, V-268  
 Philbin, James III-677  
 Pietikainen, Matti I-327  
 Pock, Thomas III-538  
 Polak, Simon II-336  
 Pollefeys, Marc II-142, III-144, IV-269,  
 IV-354, IV-368, VI-266  
 Porta, Josep M. III-370  
 Prati, Andrea VI-196  
 Preusser, Tobias V-254  
 Prinnet, Véronique IV-171  
 Pu, Jian I-257  
 Pugeault, Nicolas VI-154  
 Pundik, Dmitry III-15
- Qin, Hong III-384  
 Qing, Laiyun II-308  
 Quack, Till I-734  
 Quan, Long II-1, V-561
- Rabe, Clemens IV-582  
 Rabin, Julien V-771  
 Radke, Richard J. V-715  
 Raguram, Rahul IV-368  
 Rahtu, Esa V-366  
 Ramalingam, Srikumar III-129, V-436  
 Ramamoorthi, Ravi II-294  
 Ramanan, Deva IV-241, IV-610  
 Ramanathan, Subramanian IV-30  
 Rangarajan, Prasanna VI-405  
 Ranjbar, Mani II-580  
 Rao, Josna IV-651  
 Raptis, Michalis I-577  
 Rashtchian, Cyrus IV-15  
 Raskar, Ramesh I-86  
 Razavi, Nima I-620  
 Reid, Ian V-394  
 Reilly, Vladimir III-186, VI-252  
 Ren, Xiaofeng V-408  
 Resmerita, Elena I-185  
 Richardt, Christian III-510  
 Riemenschneider, Hayko V-29  
 Robles-Kelly, Antonio II-636

- Roca, Xavier II-678  
 Rocha, Anderson III-481  
 Rodolà, Emanuele V-519  
 Rodrigues, Rui IV-382  
 Romeiro, Fabiano I-45  
 Rosenhahn, Bodo II-128  
 Roshan Zamir, Amir IV-255  
 Roth, Stefan IV-467  
 Rother, Carsten I-143, II-465, III-272  
 Roumeliotis, Stergios I. IV-311  
 Roy-Chowdhury, Amit K. I-605  
 Rudovic, Ognjen II-350  
 Russell, Chris IV-424, V-239  
  
 Sadeghi, Mohammad Amin IV-15  
 Saenko, Kate IV-213  
 Saffari, Amir III-776, VI-29  
 Sajadi, Behzad IV-72  
 Sala, Pablo V-603  
 Salo, Mikko V-366  
 Salti, Samuele III-356  
 Salzmann, Mathieu I-677  
 Sánchez, Jorge IV-143  
 Sankar, Aditya I-341  
 Sankaranarayanan, Aswin C. I-129,  
 II-237  
 Sapiro, Guillermo V-617  
 Sapp, Benjamin II-406  
 Satkin, Scott I-536  
 Satoh, Shin'ichi I-692  
 Savarese, Silvio IV-553, V-658  
 Scharr, Hanno IV-596  
 Scheirer, Walter III-481  
 Schiele, Bernt I-720, IV-467, VI-182  
 Schindler, Konrad I-466, IV-467, VI-182  
 Schmid, Cordelia I-522, I-634  
 Schmidt, Stefan III-735  
 Schnörr, Christoph II-494, III-735  
 Schofield, Andrew J. IV-58  
 Schroff, Florian IV-438  
 Schuchert, Tobias IV-596  
 Schwartz, William Robson VI-476  
 Sclaroff, Stan I-494, III-453  
 Sebe, Nicu IV-30  
 Seitz, Steven M. I-341, II-29  
 Seo, Yongduek IV-115  
 Serradell, Eduard III-58  
 Shah, Mubarak III-186, III-439,  
 IV-255, VI-252  
  
 Shan, Qi VI-364  
 Shan, Shiguang I-327, II-308  
 Shapiro, Linda G. III-621  
 Sharma, Avinash V-743  
 Shashua, Amnon II-336  
 Shechtman, Eli I-341, III-29  
 Sheikh, Yaser III-158  
 Shen, Chunhua II-608  
 Shen, Xiaohui I-411  
 Shetty, Sanketh V-644  
 Shi, Yonggang III-594  
 Shih, Jonathan I-663  
 Shiratori, Takaaki III-158  
 Shoaib, Muhammad II-128  
 Shu, Xianbiao VI-393  
 Siegwart, Roland V-296  
 Sigal, Leonid III-243  
 Silva, Jorge G. III-172  
 Singh, Vivek Kumar III-314  
 Sivalingam, Ravishankar IV-722  
 Sivic, Josef I-748, III-677  
 Sminchisescu, Cristian II-480  
 Smith, William A.P. VI-112  
 Snavely, Noah II-29, II-791  
 Soatto, Stefano I-577, III-692  
 Solmaz, Berkan VI-252  
 Sommer, Stefan I-425, VI-43  
 Song, Bi I-605  
 Song, Mingli V-631  
 Song, Yi-Zhe IV-694  
 Spera, Mauro II-378  
 Spinello, Luciano V-296  
 Stalder, Severin I-369  
 Staudt, Elliot I-605  
 Stevenson, Suzanne V-183  
 Stoll, Carsten IV-568  
 Strecha, Christoph IV-778  
 Sturgess, Paul IV-424  
 Sturm, Peter II-85  
 Su, Guangda IV-44  
 Su, Zhixun I-115  
 Sukthankar, Rahul I-508  
 Sun, Jian I-1  
 Sun, Ju III-748  
 Sun, Min V-658  
 Sundaram, Narayanan I-438  
 Sunkavalli, Kalyan II-251  
 Suppa, Michael II-183  
 Suter, David V-533  
 Szeliski, Richard II-29

- Sznaier, Mario II-71  
 Szummer, Martin I-776
- Ta, Vinh-Thong IV-638  
 Taguchi, Yuichi III-129, V-436  
 Tai, Xue-Cheng VI-379  
 Tai, Yu-Wing VI-323  
 Takahashi, Keita IV-340  
 Tan, Ping II-265  
 Tan, Xiaoyang VI-504  
 Tang, Feng III-258  
 Tang, Hao VI-490  
 Tang, Xiaou I-1, VI-420  
 Tanskanen, Petri IV-269  
 Tao, Hai III-258  
 Tao, Linmi IV-525  
 Tao, Michael W. I-31  
 Taskar, Ben II-406  
 Taylor, Graham W. VI-140  
 Theobalt, Christian IV-568  
 Thompson, Paul III-594  
 Tian, Tai-Peng III-453  
 Tighe, Joseph V-352  
 Tingdahl, David I-734  
 Todorovic, Sinisa II-721, V-57  
 Toldo, Roberto V-589  
 Tomasi, Carlo III-663  
 Tombari, Federico III-356  
 Tong, Yan I-354  
 Torii, Akihiko II-100  
 Torr, Philip H.S. IV-424, V-239  
 Torralba, Antonio I-762, II-707, V-85  
 Torresani, Lorenzo I-776  
 Torsello, Andrea V-519  
 Tosato, Diego II-378  
 Toshev, Alexander II-406  
 Tran, Duan IV-227  
 Traver, V. Javier IV-680  
 Tretiak, Elena II-57  
 Triebel, Rudolph V-296  
 Troje, Nikolaus F. III-243  
 Tsang, Ivor Wai-Hung IV-1  
 Tu, Peter H. I-354  
 Tu, Zhuowen III-328, V-15  
 Turaga, Pavan III-286  
 Turaga, Pavan K. I-129  
 Turek, Matthew I-549  
 Turek, Matthew W. II-664  
 Tuzel, Oncel II-237, V-436
- Urtasun, Raquel I-677
- Valgaerts, Levi IV-568  
 Valmadre, Jack III-467  
 Van Gool, Luc I-143, I-369, I-452,  
 I-620, I-734, III-425  
 Vasquez, Dizan V-296  
 Vasudevan, Ram III-101  
 Vazquez-Reina, Amelio V-268  
 Veeraraghavan, Ashok I-100, II-237  
 Veksler, Olga V-211  
 Verbeek, Jakob I-634  
 Vese, Luminita I-185  
 Vicente, Sara II-465  
 Villanueva, Juan J. VI-280  
 Vondrick, Carl IV-610  
 von Lavante, Etienne V-743  
 Vu, Ngoc-Son I-313
- Wah, Catherine IV-438  
 Walk, Stefan VI-182  
 Wang, Bo III-328, V-15  
 Wang, Chen I-257  
 Wang, Gang V-169  
 Wang, Hua III-762, IV-793, VI-126  
 Wang, Huayan II-435, IV-497  
 Wang, Jue V-617  
 Wang, Kai I-591  
 Wang, Lei III-524  
 Wang, Liang I-257, IV-708  
 Wang, Peng II-608  
 Wang, Qifan IV-525  
 Wang, Shengnan IV-87  
 Wang, Xiaogang VI-420  
 Wang, Xiaosong V-478  
 Wang, Xiaoyu III-200  
 Wang, Xinggang III-328, V-15  
 Wang, Yang II-580, V-155  
 Wang, Zengfu V-729  
 Wang, Zhengxiang I-706  
 Watanabe, Takuya VI-337  
 Wedel, Andreas IV-582  
 Weickert, Joachim IV-568  
 Weinland, Daniel III-635  
 Weiss, Yair I-762  
 Welinder, Peter IV-438  
 Werman, Michael II-749  
 Wheeler, Frederick W. I-354  
 Wilburn, Bennett I-59  
 Wildes, Richard I-563, IV-511

- Wojek, Christian IV-467  
 Wong, Tien-Tsin V-422  
 Wu, Changchang II-142, IV-368  
 Wu, Jianxin II-552  
 Wu, Szu-Wei I-271  
 Wu, Xiaolin VI-351  
 Wu, Ying I-411, I-480  
 Wyatt, Jeremy L. IV-58
- Xavier, João IV-283  
 Xia, Yan V-729  
 Xiao, Jianxiong V-561  
 Xie, Xianghua IV-680  
 Xing, Eric P. V-85, V-785  
 Xu, Bing-Xin V-658  
 Xu, Li I-157  
 Xu, Wei VI-434
- Yamato, Junji III-215  
 Yan, Junchi III-790  
 Yan, Shuicheng III-748  
 Yang, Jianchao III-566, V-113  
 Yang, Jie III-790  
 Yang, Lin IV-624  
 Yang, Meng VI-448  
 Yang, Qingxiong IV-87  
 Yang, Ruigang IV-708  
 Yang, Xingwei III-411, V-450, V-757  
 Yang, Yezhou V-631  
 Yao, Angela III-425  
 Yarlagadda, Pradeep V-197  
 Yau, Shing-Tung V-672  
 Yeh, Tom II-693  
 Yezzi, Anthony I-73, VI-71  
 Yilmaz, Alper IV-410  
 Young, Peter IV-15  
 Yu, Chanki IV-115  
 Yu, Jin V-533  
 Yu, Jingyi I-15  
 Yu, Kai VI-434, V-113, V-141  
 Yu, Xiaodong V-127
- Yuan, Jing VI-379  
 Yuan, Xiaoru I-257  
 Yuen, Jenny II-707  
 Yuille, Alan IV-539, V-43
- Zach, Christopher IV-354  
 Zaharescu, Andrei I-563  
 Zeng, Wei V-672  
 Zeng, Zhi VI-532  
 Zhang, Cha III-229  
 Zhang, Chenxi IV-708  
 Zhang, Guofeng V-422  
 Zhang, Haichao III-566  
 Zhang, Honghui V-561  
 Zhang, Junping I-257  
 Zhang, Lei IV-157, VI-448, V-506  
 Zhang, Shaoting III-607  
 Zhang, Tong V-141  
 Zhang, Wei I-115, VI-420  
 Zhang, Yanning III-566  
 Zhang, Yuhang III-524  
 Zhang, Zhengyou III-229  
 Zhao, Bin V-785  
 Zhao, Mingtian IV-101  
 Zhao, Qiping IV-101  
 Zhao, Yong VI-351  
 Zheng, Ke Colin III-621  
 Zheng, Wenming VI-490  
 Zheng, Ying III-663  
 Zhou, Changyin VI-337  
 Zhou, Jun II-636  
 Zhou, Qian-Yi III-115  
 Zhou, Xi V-141  
 Zhou, Yue III-790  
 Zhou, Zhenglong II-265  
 Zhu, Long (Leo) V-43  
 Zhu, Song-Chun IV-101  
 Zickler, Todd I-45, II-251  
 Zimmer, Henning IV-568  
 Zisserman, Andrew III-677  
 Zitnick, C. Lawrence II-170