Kostas Daniilidis
Petros Maragos
Nikos Paragios (Ed

LNCS 6312

Compute
ECCV 201

11th European Conference
Heraklion, Crete, Greece, S
Proceedings, Part II

Lecture Notes in Computer Science 6312

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

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 II

Springer

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 3-642-15551-0 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-15551-2 Springer Berlin Heidelberg New York

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.

September 2010

Kostas Daniilidis
Petros Maragos
Nikos Paragios

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

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

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

Luo, Jiebo
Lyu, Siwei
Ma, Xiaoxu
Mairal, Julien
Maire, Michael
Maji, Subhransu
Maki, Atsuto
Makris, Dimitrios
Malisiewicz, Tomasz
Mallick, Satya
Manduchi, Roberto
Manmatha, R.
Marchand, Eric
Marcialis, Gian
Marks, Tim
Marszalek, Marcin
Martinec, Daniel
Martinez, Aleix
Matei, Bogdan
Mateus, Diana
Matsushita, Yasuyuki
Matthews, Iain
Maxwell, Bruce
Maybank, Stephen
Mayer, Helmut
McCloskey, Scott
McKenna, Stephen
Medioni, Gerard
Meer, Peter
Mei, Christopher
Michael, Nicholas
Micusik, Branislav
Minh, Nguyen
Mirmehdi, Majid
Mittal, Anurag
Miyazaki, Daisuke
Monasse, Pascal
Mordohai, Philippos
Moreno-Noguer, Francesc
Mori, Greg
Morimoto, Carlos
Morse, Bryan
Moses, Yael
Mueller, Henning
Mukaigawa, Yasuhiro
Mulligan, Jane
Munich, Mario
Murino, Vittorio
Namboodiri, Vinay
Narasimhan, Srinivasa
Narayanan, P.J.
Naroditsky, Oleg
Neumann, Jan
Nevatia, Ram
Nicolls, Fred
Niebles, Juan Carlos
Nielsen, Mads
Nishino, Ko
Nixon, Mark
Nowozin, Sebastian
O'donnell, Thomas
Obozinski, Guillaume
Odobez, Jean-Marc
Odone, Francesca
Ofek, Eyal
Ogale, Abhijit
Okabe, Takahiro
Okatani, Takayuki
Okuma, Kenji
Olson, Clark
Olsson, Carl
Ommer, Bjorn
Osadchy, Margarita
Overgaard, Niels Christian
Ozuysal, Mustafa
Pajdla, Tomas
Panagopoulos, Alexandros
Pandharkar, Rohit
Pankanti, Sharath
Pantic, Maja
Papadopoulo, Theo
Parameswaran, Vasu
Parikh, Devi
Paris, Sylvain
Patow, Gustavo
Patras, Ioannis
Pavlovic, Vladimir
Peleg, Shmuel
Perera, A.G. Amitha
Perronnin, Florent
Petrou, Maria
Petrovic, Vladimir
Peursum, Patrick
Philbin, James
Piater, Justus
Pietikainen, Matti
Pinz, Axel
Pless, Robert
Pock, Thomas
Poh, Norman
Pollefeys, Marc
Ponce, Jean
Pons, Jean-Philippe
Potetz, Brian
Prabhakar, Salil
Qian, Gang
Quattoni, Ariadna
Radeva, Petia
Radke, Richard
Rakotomamonjy, Alain
Ramanan, Deva
Ramanathan, Narayanan
Ranzato, Marc'Aurelio
Raviv, Dan
Reid, Ian
Reitmayr, Gerhard
Ren, Xiaofeng
Rittscher, Jens
Rogez, Gregory
Rosales, Romer
Rosenberg, Charles
Rosenhahn, Bodo
Rosman, Guy
Ross, Arun
Roth, Peter
Rother, Carsten
Rothganger, Fred
Rougon, Nicolas
Roy, Sebastien
Rueckert, Daniel
Ruether, Matthias
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
Singaraju, Dheeraj
Singh, Maneesh
Singh, Vikas
Sinha, Sudipta
Sivic, Josef
Slabaugh, Greg
Smeulders, Arnold
Sminchisescu, Cristian
Smith, Kevin
Smith, William
Snavely, 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
Süsstrunk, 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
Wildes, Richard
Williams, Oliver
Wills, Josh
Wilson, Kevin
Wojek, Christian
Wolf, Lior
Wright, John
Wu, Tai-Pang
Wu, Ying
Xiao, Jiangjian
Xiao, Jianxiong
Xiao, Jing
Yagi, Yasushi
Yan, Shuicheng
Yang, Fei
Yang, Jie
Yang, Ming-Hsuan
Yang, Peng
Yang, Qingxiong
Yang, Ruigang
Ye, Jieping
Yeung, Dit-Yan
Yezzi, Anthony
Yilmaz, Alper
Yin, Lijun
Yoon, Kuk Jin
Yu, Jingyi
Yu, Kai
Yu, Qian
Yu, Stella
Yuille, Alan
Zach, Christopher
Zaid, Harchaoui
Zelnik-Manor, Lihi
Zeng, Gang
Zhang, Cha
Zhang, Li
Zhang, Sheng
Zhang, Weiwei
Zhang, Wenchao
Zhao, Wenyi
Zheng, Yuanjie
Zhou, Jinghao
Zhou, Kevin
Zhu, Leo
Zhu, Song-Chun
Zhu, Ying
Zickler, Todd
Zikic, Darko
Zisserman, Andrew
Zitnick, Larry
Zivny, Stanislav
Zuffi, Silvia

Sponsoring Institutions

Platinum Sponsor

Gold Sponsors

Silver Sponsors

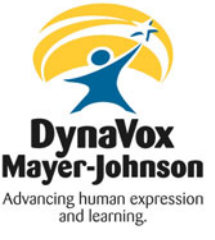


Table of Contents – Part II

Spotlights and Posters M2

Humans and Faces

Spotlights and Posters T1

Resampling Structure from Motion

Tian Fang and Long Quan

The Hong Kong University of Science and Technology,
Clear Water Bay, Kowloon, Hong Kong, China
{fangtian,quan}@cse.ust.hk

Abstract. This paper proposes a hierarchical framework that resamples 3D reconstructed points to reduce computation cost on time and memory for very large-scale Structure from Motion. The goal is to maintain accuracy and stability similar for different resample rates. We consider this problem in a level-of-detail perspective, from a very large scale global and sparse bundle adjustment to a very detailed and local dense optimization. The dense matching are resampled by exploring the redundancy using local invariant properties, while 3D points are resampled by exploring the redundancy using their covariance and their distribution in both 3D and image space. Detailed experiments on our resample framework are provided. We also demonstrate the proposed framework on large-scale examples. The results show that the proposed resample scheme can produce a 3D reconstruction with the stability similar to quasi dense methods, while the problem size is as neat as sparse methods.

1 Introduction

Nowadays growing demands on realtime mapping and localization, large scale digital city modeling [1] push the scale of Structure from Motion (SfM) [2] to the limits of our computing capacity again and again. The pipeline of the SfM follows a divide-conquer-merge methodology. The collected images are first processed to extract features independently. Then a matching and elementary reconstruction process, e.g. projective reconstruction, is carried out to solve the SfM in pairwise or triplet manner. Such pairwise and triplet reconstruction are the fundamental building blocks (sub-problem) of any SfM system. The sub-problems are merged into a consistent and complete result using a hierarchical [3] or incremental [4] merging process. To ensure consistency across the merged sub-problems, a golden standard method–bundle adjustment [5], is used. Unfortunately, like any other problems solved by divide-conquer-merge methodology, the huge merged problem will exhaust the computation resource. In structure from motion, it is challenging to fit the large scale bundle adjustment problem into memory, which is an initial motivation to our work.

To reduce the problem size, a common approach is to explore the redundancy. Lhuillier et al. [6] proposed a resample scheme for dense matching. The local resample scheme not only reduces the consumption of the computation resource due to large amount of pixel wise matches, but also improves the reliability of

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 1–14, 2010.

resampled matching by using local-plane-model validation. To reduce the redundancy in-between images, key-frames are extracted [7] given sequential input images, so the computation can focus on the reduced set of images. Meanwhile, with unordered images, Snavely et al. [8] proposed a skeleton representation of the dominant cameras which are then used as the foundation to speed up following incremental camera insertions and 3D point reconstruction.

Decoupling is another strategy to tackle large scale problems. Ni et al. [9] partitioned the large scale problem into overlapping blocks that fit to main memory and bundle each block respectively in an iterative inter-partition refinement manner. However, due to high inter-connectivity between the parameters, it is difficult to construct a pure independent partition from the original scene.

It is also another compromise to constrain the problem being solved only locally. Local bundle adjustment [10,11] is proposed to use only the images and features in the last few images in image sequences in the bundle adjustment instead of using all images and features.

In contrast to finding redundancy in cameras, in this paper, from a level-of-detail perspective, we propose a hierarchical resampling framework on 3D points for the large scale SfM, which fits the large scale problem into main memory. Moreover, with the concept of resampling, we set up a full picture of the spectrum of level-of-detail (multi-scale) for geometry reconstruction (Figure 1). In this spectrum, a very dense local reconstruction, e.g. multi-view stereo [12], can transit to a semi-dense reconstruction [6], which can later be resampled to a sparse reconstruction. This transition is also valid vice versa.

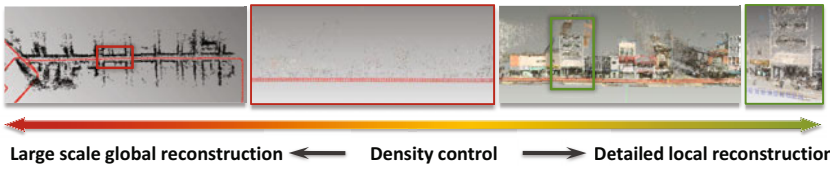


Fig. 1. Spectrum of the level-of-detail of the bundle adjustment

In this paper, we first review the basic notation and background knowledge of bundle adjustment in Section 2. Next, our hierarchical resampling scheme is introduced in Section 3. After that, an approximate bundle adjustment method and an out-of-core merging process are introduced in Section 3.3 based on the resampling scheme. The experiments and discussion are given in Section 4. Finally, we conclude our work in Section 5.

2 Short Review of Bundle Adjustment

Given a set of input images $\hat{I} = \{I_0, \ldots, I_j\}$, let $\mathbf{c} = \{\mathbf{c}_0{}^\mathsf{T}, \ldots, \mathbf{c}_j{}^\mathsf{T}\}$ be the parameter vectors of all cameras $\hat{C}$ associated with $\hat{I}$ and $\mathbf{p} = \{\mathbf{p}_0{}^\mathsf{T}, \ldots, \mathbf{p}_i{}^\mathsf{T}\}$ be the parameter vectors of all 3D points $\hat{P} = \{P_0, \ldots, P_i\}$. A visibility function

$V(i,j)$ is defined to be 1 when P_i is visible in I_j. Otherwise, $V(i,j)$ is defined to be 0. Then a classic bundle adjustment problem can be expressed as a nonlinear least square problem:

$$[\mathbf{c}\ \mathbf{p}]^{\mathsf{T}} = \arg\min_{\mathbf{c},\mathbf{p}} \|\mathbf{x} - f(\mathbf{c},\mathbf{p})\|^2 \tag{1}$$

Equation 1 is to estimate $\mathbf{c}$ and $\mathbf{p}$ that optimize the re-projection error, given the set of projections $\hat{X} = \{X_{ij} | \forall i,j \text{ where } V(i,j)\ is\ 1\}$ of 3D points onto input images. We also use $|V|$ to denote the number of projections in images. $\mathbf{x}$ is the concatenation of the column vector of all projections $\{\mathbf{x}_{ij}^T | \forall i,j \text{ where } V(i,j)\ is\ 1\}^T$. $f(\cdot)$ is the model of projection. The variance of the estimation can also be estimated using the inverse of the Hessien of Equation 1, i.e. $\mathbf{H}^{-1}$ according to the perturbation analysis [13]. However, due to the gauge freedom, the estimation of $\mathbf{c}$, $\mathbf{p}$ and their covariance are up to the choice of the gauge. The estimations of $\mathbf{c}$ and $\mathbf{p}$ that yield the same optimized value for Equation 1 form a manifold called gauge orbit. In order to obtain a unique estimation of $\mathbf{c}$ and $\mathbf{p}$, additional constrains $\mathcal{C}$ on $\mathbf{c}$ and $\mathbf{p}$ are required. This process is called gauge fixing. The covariance of $\mathbf{c}$ and $\mathbf{p}$ is highly related to the choice of gauge as well. However, Morris has shown that this set of numerically unequal covariance is essentially equivalent geometrically to normal covariance [13]. This fact makes the normal covariance become an unified criteria for the quality of an estimation.

3 Hierarchical Resampling

Our framework of hierarchical resampling starts from the resampling of dense matching and moves to the resampling of 3D points. The goal of the resampling is to simplify the large scale problem so that the problem can be solved efficiently, while maintaining the stability of the reconstruction.

3.1 Dense Matching Resample

Thanks to the robustness of rotation and scale invariance features [14], sparse reconstruction is quite popular nowadays. However, as demonstrated in [6] and in later experiments, unbalance sparse features in image can make the geometry reconstruction problematic. Hence, matching propagation is still recommended to maximize the stability of SfM. However, it overwhelms the computer to involve all the propagated pixel matches. Therefore, we use the resampling strategy proposed in [6] to resample the semi-dense pixel matches. The general process proceeds as following steps.

(1) Pixels are aggregated into local groups. The local group should be small enough so that the pixels in the same group share some invariance, e.g. local plane assumption. On the other hand, the local group should be also large enough to contain enough reliable observations. For simplicity, regular 8 by 8 pixels square grids are used in this implementation. Over-segmentation algorithm [15] that could generate equal-size and edge sensitive over-segmentation is also a good candidate for partitioning pixels into local groups.

(2) Local groups of pixel correspondences are evaluated using some local invariance property. Only the groups that pass the evaluation will be kept and a representative point correspondence will be generated for later stages. In this implementation, we used local affine transform as the invariance property in a local group.

Please note that the local invariance hypothesis, i.e. local affine transform, does not need any knowledge of the global motion between two images. Hence, this step can be used before any 2-view or 3-view geometry reconstruction to remove redundant information.

3.2 3D Points Resample

After local geometries are estimated, these local geometries are further merged into a global geometry. Because the number of resampled semi-dense matches is usually $10 \sim 1000$ times more than the number of sparse matches, the memory runs out fast if all matches are used. In order to maintain the problem solvable in main memory, we need to resample 3D points to reduce the problem size. At the same time, we need to keep in mind that removing the redundant 3D points should not harm the optimization itself. Hence, we should first figure out which kind of points are less useful for bundle adjustment.

The meaning of "less useful" is twofold. First, some points themselves are poorly reconstructed. Geometrically, small base line and small angle between the reprojected rays for triangulation yields poor estimation of the 3D points. Mathematically, the badness of the estimation of 3D points can be expressed as the covariance of the estimated parameters using perturbation analysis, but this covariance is highly related to the choice of gauge. As reviewed in Section 2, normal covariance can be used to represent this set of geometrically equivalent covariance regardless of the choice of gauge. More concretely, we take the diagonal blocks of the normal covariance matrix corresponding to the parameters of 3D points. Then each 3×3 covariance matrix is interpreted as an uncertainty ellipsoid. The sum of the principle axes of an uncertainty ellipsoid is taken as the measurement of the uncertainty of a 3D point.

Second, the removed 3D points should not in turn harm the estimation of the parameters of cameras. Remaining points should span the whole reconstructed scene and distribute uniformly in both 3D and image space. These uniform points make the residual of Equation 1 distributed well over all points and make the estimation of camera parameters well constrained. This uniformness in 3D and image space, can be measured with the density of points in 3D and image space.

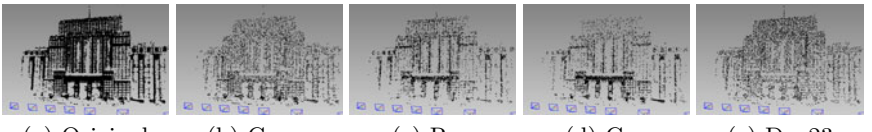
Therefore,for each point P_i, we define a score to measure its redundancy as:

$$s_i = u_i \cdot \rho_i \cdot \min_{\forall j \ V(i,j) \text{ is } 1} \rho^i_{I_j} \tag{2}$$

where u_i is the uncertainty of P_i, ρ_i is the density of 3D points around point P_i, and $\rho^i_{I_j}$ is the density of the 2D projections in image I_j where P_i is visible. Points with higher scores are regarded as more redundant and less useful than points with lower scores.

Now, we can at least remove a point with the highest score each time to resample points. Unfortunately, the scores of the remaining points change when any 3D points are removed. It is not computationally practical to re-compute the scores for the remaining points every time a point is removed, as simply the covariance computation takes $O(|V||\hat{C}| + |\hat{P}|r^2 + |\hat{C}|^3)$, given r is the maximal number of projections a 3D point has. In contrast to this greedy strategy, we tackle this problem using a stochastic sampling process, which only require $O(|\hat{P}|)$ time, given a precomputed score of each point. The sampling process can be interpreted as the higher score a point has, the more likely it should be removed. In the stochastic sampling process, s_i is first computed for each point. Then we can *select the 3D points to be removed* proportionally to this score using SUS (stochastic universal sampling) [16]. To build the sampling distribution used in SUS, we normalize the scores of all 3D points by their sum. With this resample scheme, we can define a *downsample ratio* (*resample rate*) as the ratio of the number of remaining points to the number of original points.

Figure 2 shows how the terms in Equation 2 affect the resampled 3D points. The full reconstruction in (a) is generated using the semi-dense reconstruction [6] [1]. The results show that the random sample gives a resampled result of similar distribution to the original reconstruction, where the resampled points cluster around textured region. If only the covariance is considered in the score, the remaining points are clustered around the places of better geometrical condition, especially where is close to cameras. "Den23" consisting of only 2D and 3D density makes resampled points uniformly distributed. For more experimental analysis, please refer to Section 4.



(a) Original (b) Comp (c) Ran (d) Cov (e) Den23

Fig. 2. (a) Original result with all reconstructed points. 10% points are kept by resampling using our score function (b) Comp, using random sample (c) Ran, using only the uncertainty measure (d) Cov, using the combination of 3D density and 2D density (e) Den23. These abbreviations have the same meaning as here throughout this paper.

3.3 Approximate Bundle Adjustment and Out-of-Core Hierarchical Merging

Next, we use the above resample strategy to speed up the bundle adjustment and to adapter the original hierarchical merge process into an out-of-core manner.

Approximate bundle adjustment. We would like to use the resampled geometry to approximate the bundle adjustment. First a full bundle adjustment problem BA_0 is resampled into a simplified bundle adjustment problem BA_s according

[1] Reconstructed with 10 input images at resolution 2400×1600 pixels which is different from the resolution used to generate the results for the same scene in Section 4.

to the capacity of main memory. Then a bundle adjustment of both motion and structure parameters is carried out on BA_s. The 3D points $\mathbf{P}_c = \{P_i \in BA_0 \text{ and } \notin BA_s\}$ are in turn estimated using the optimized parameters in BA_s by e.g. linear triangulation. Finally such $\mathbf{P}_c$ will be optimized with a bundle adjustment only on structures. This process can be iterated several times. Each time, the original BA_0 is resampled again according to latest updated parameters. However, we found that this process usually converges in an iteration with our resample scheme. Hence, it is much faster than solving a full bundle adjustment problem.

Out-of-core merging. We can also adapt the hierarchical merge process [3] to an out-of-core manner based on our resample scheme. Given sequential images, the local triplet geometries are first reconstructed for every consecutive 3 images using the semi-dense correspondences that are resampled from dense propagated matching. Then we hierarchically merge the local geometries into a global geometry. The merging process starts from finding a transformation, e.g. similarity transformation, which aligns the overlapping cameras between two consecutive local geometries. We merge the overlapping cameras by keeping either one of them. Then the points from different local geometries are merged if they have overlapping projections. In our implementation, we use 0.3 pixels as the threshold for overlapping projections. Finally, bundle adjustment is applied on the merged geometry to obtain higher level local geometries.

The above process can be carried out in an out-of-core manner as following. Given two local geometries G_0 and G_1, if the bundle adjustment on the merged geometries G_{01} does not fit into main memory, G_0 and G_1 will be resampled to GS_0 and GS_1, which are merged into GS_{01}. Only the simplified geometries are used in further merging and bundle adjustment, while the removed 3D points are dumped to the hard disk. The resample rate is controlled by the bound of memory available for a program. In the end, we obtain an optimized resampled global geometry. As the number of levels of the hierarchical merging is $O(\log n)$ given n local geometries, the total IO required is bounded by $O(n \log n)$. Therefore, this process is I/O efficient.

4 Experiment and Discussion

In this section, we first describe the implementation of our system. Then the proposed resample scheme is validated on moderate-scale data sets and large-scale data sets, followed with the discussion.

4.1 Implementation

Our SfM pipeline follows the hierarchical strategy and is in calibrated framework. SIFT or SURF can be used as sparse features. The matching propagation algorithm [17,18] is implemented. ANN is used for the approximate nearest neighbor searching. We use the calibrated 5 points algorithm [19] to reconstruct 2-view and 3-view geometries. The bundle adjustment is handled by SBA [20].

Fig. 3. Typical input images. From left to right, Hall of Prayers (HALL), BUILDING, OXFORD, Canton and UNC.

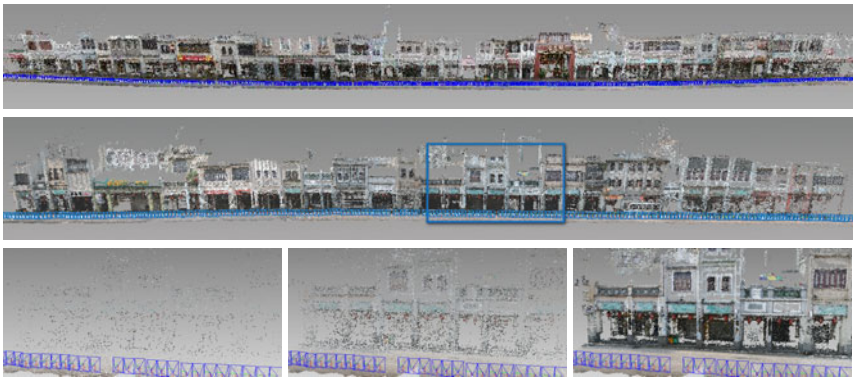


Fig. 4. Canton sequence. Top row is the complete reconstruction of Canton#1 with resampled 3D points. Middle row is the complete reconstruction of Canton#2 with resampled 3D points. Bottom row is a close up view of the blue rectangular region of the middle row in three different resample rates 5%, 20% and 100%.

For computing scores, a fast covariance computation [6] is used to obtain the normal covariance of the position vectors of reconstructed cameras and 3D points. We approximate the density of points around a point by counting the number of points inside a fixed radius neighborhood around a point. This range search is also speeded up by ANN. To find the radius that is used to compute 3D density, we first find the distance of each point in $\hat{\mathbf{P}}$ to its nearest neighbor. Then the average distance d_{av} of the first 50% are computed and set as the searching radius for 3D density. For 2D density, 8 pixels is used as the radius to compute the density.

In SUS, one thing has to be noted is that the samples are allowed to be re-drawn, so a few points with higher scores may be selected multiple times. This behavior is normal in the sense of statistic, but it is not acceptable in our system, because the number of points to be removed is strictly bounded by the capacity of the main memory. To overcome this problem, we run SUS iteratively on the points that are not yet selected in the past iterations until enough points are selected. In our experiment, the points of required number can usually be selected in 2 or 3 iterations.

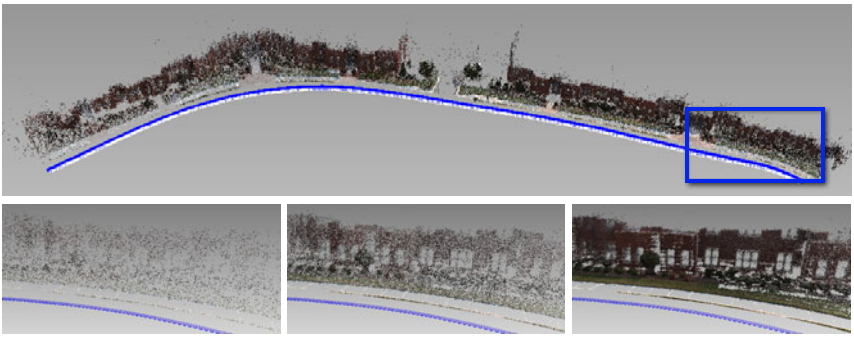


Fig. 5. UNC sequence. Top row is the complete reconstruction with resampled 3D points. Bottom row shows close-up views of the blue rectangular region of the top row in three different resample rates 5%, 20% and 100%.

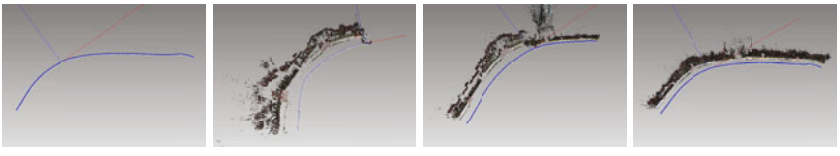


Fig. 6. Comparison of UNC sequence

4.2 Moderate Scale Data Sets

Here we have three moderate scale data sets, which are denoted as capitalized HALL, BUILDING and OXFORD. The typical input images are shown in Figure 3. These three examples represent three types of typical camera motions, moving circularly with viewing direction perpendicular to the moving direction, moving in a straight line while the camera focusing on a center object, and moving along the viewing direction. HALL was taken while the photographer moved along a circular path around the center object. BUILDING was taken while the photographer followed a straight line on the ground.

We reconstruct these examples using both sparse and semi-dense matches. The SfM pipeline is the same, only the matches are different. We do not involve any prior knowledge of camera motion, e.g. loop constrain or straight line movement. In HALL, SURF features are used, while in the other examples, SIFT features are used as sparse features. In Figure 7, the semi-dense reconstruction has superior quality in both HALL and BUILDING, thanks to the extra and more balance propagated matches. In HALL, the sparse reconstruction cannot close the loop, while the positions of the cameras at both ends in BUILDING are bended forward in the sparse reconstruction. In OXFORD, sparse method and semi-dense method produce similar results.

Then we gradually resampled the merged reconstruction in the "MERGE" column of Figure 7 with a resample rate from 100% to 0.2% using different scores

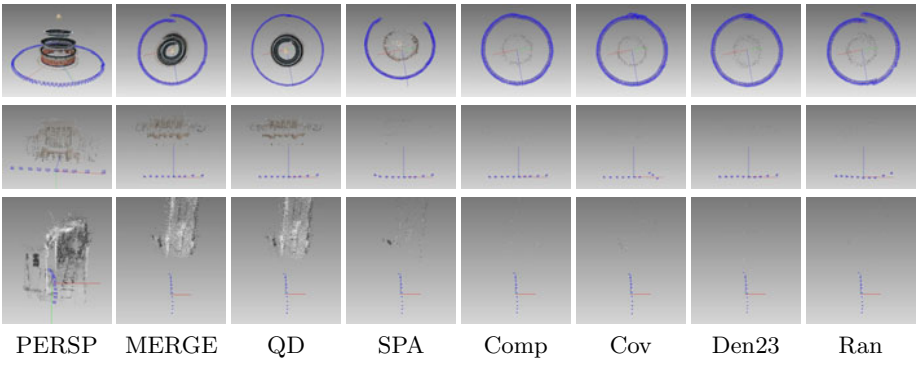


Fig. 7. The reconstruction and resampling of three moderate scale data sets. From top to bottom, they are the results for HALL, BUILDING, and OXFORD respectively. "PERSP" shows an perspective view of the semi-dense reconstruction. "MERGE" shows the merged semi-dense reconstruction before bundle adjustment. "QD" shows the bundle adjustment with all points in "MERGE". "SPA" shows the results of sparse reconstruction. "Comp", "Cov", "Den23" and "Ran" show the results of bundle adjustment on a resampled point set from "MERGE" at the ratio in the "md" column of Table 1. Please refer to Table 1 for some statistic.

and bundled the resampled geometries. After the visual inspection of the results, we list the minimal resample rate which still yields reasonable reconstruction in "md" column of Table 1. The visual quality of the reconstruction can be found in the right most 4 columns of Figure 7.

In HALL and BUILDING, "Den23" can maintain the overall trajectory of camera motions, as it keeps the points evenly in 3D and image space, which makes the estimation of camera well constrained. However, "Den23" does not consider the quality of the 3D points. Usually, 3D points with poor quality can make the optimization bias. We can observe this small bias in the "Den23" of HALL. The top of the trajectory is slightly bended inwards to the center of circle compared to "QD". At the same time, only relying on the uncertainty terms also induces bias in reconstruction due to the unbalanced resampled points. "Cov" of BUILDING is an example. The right most three cameras were not recovered correctly. "Cov" of HALL can be reconstructed well because the scene and the camera motion are both symmetric. After removing a few highly uncertain points at the background, the remaining points have similar uncertainty. Hence, resampling HALL with "Cov" still yields a quite uniform point distribution. In both of HALL and BUILDING, "Ran" gives bad results, while "Comp" which combines the strength of "Cov" and "Den23" produces a reconstruction better than sparse reconstruction with even fewer points. The results also show that it is worth spending effort on carefully selecting 3D points. In OXFORD, visually, it is hard to tell the difference between the reconstruction using different resample score. This will be explained in the following analysis of Figure 8.

We also plot the average covariance of the position of cameras and the average covariance of the 3D points in Figure 8. In HALL and BUILDING, "Comp" gets

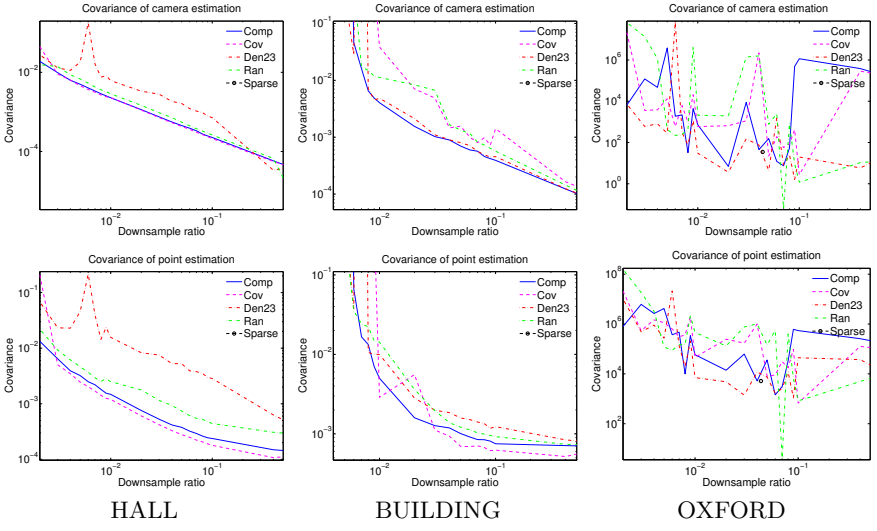


Fig. 8. Top row: the covariance of the position of cameras after bundle adjustment on the resampled point set vs the downsample ratio. Bottom row: the covariance of the position of 3D points after bundle adjustment on the resampled point set vs the downsample ratio. In HALL and BUILDING, the covariance of the sparse reconstruction is too large to be plotted in the figure region. Please refer to Table 1 for where the bundle adjustment of the resampled point set fails.

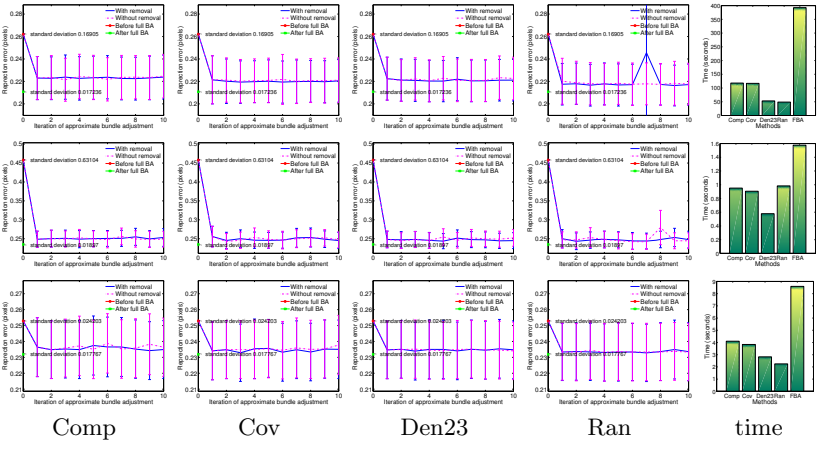


Fig. 9. The left 4 columns: the reprojection error and its standard deviation after each iteration of approximate bundle adjustment using different scores. "With removal" means after each iteration, the points with reprojection error larger than 2 pixels are removed. "Without removal" means nothing is removed after each iteration. "Full BA" stands for full bundle adjustment with all the points and cameras. The right most column: the runtime comparison. From top to bottom, they are the results for HALL, BUILDING, and OXFORD respectively.

the estimation of cameras with lowest uncertainty. However, it is not surprising that "Comp" does not always perform best for the estimation of 3D points. The reason is that given a reliable camera reconstruction, removing the points with high uncertainty greedily decreases the average covariance of 3D points most. This is what "Cov" tries to do. However, if the remaining points cannot produce reliable camera estimation, the removal of the points will harm the estimation of remaining points. This is why "Cov" cannot perform better than "Comp" in average covariance of 3D points when the downsample ratio is small. In OXFORD, the estimated covariance just jumps up and down almost randomly, because the motion of camera in OXFORD is one of the typical degenerated case of covariance estimation. However, our resample score still resists to this problematic covariance estimation and gets results not worse than uniform random resample.

Finally, we carried out an experiment on the approximate bundle adjustment. HALL, BUILDING and OXFORD are resampled at downsample ratio 0.02, 0.05 and 0.1 respectively. The average reprojection error and its standard deviation are plotted in Figure 9. The reprojection error and standard deviation before and after full bundle adjustment involving all 3D points and cameras are also plotted as baselines. We can observe that with either resample score, approximate bundle adjustment can optimize Equation 1 to almost the same residual error level as full bundle adjustment in only one iteration. However, in HALL and BUILDING, "Ran" gives us a bumping reprojection error and standard deviation after a few iterations, because "Ran" resamples points uniformly without any guidance. Sometimes "Ran" just picks up a set of points that is bad for bundle adjustment. In contrast, "Comp", "Cov", and "Den23" give better stability after a few iteration, because they resample points according to some robust criteria. In OXFORD, it is not surprising that different strategies just perform similarly, given the perturbed covariance estimation in Figure 8.

In the right most column of Figure 9, we compare the time of the first iteration of our approximate bundle adjustment and the full bundle adjustment. The time of the approximate bundle adjustment includes the time for computing scores. "Ran" is the fastest, as it does not require any computation on scores. "Den23" is second fastest, because the computation of density is moderate compared to the computation of normal covariance. The running time of "Comp" and "Cov" is similar, because the computation of normal covariance dominates the running time compared to the computation of density. However, an exceptional case is BUILDING, where "Ran" runs slowest. The reason is that although other methods spend more time on computing scores, they converge fast in optimization because of better resampling.

4.3 Large Scale Data Sets

In the experiment for large scale data set, the memory bound is manually set at 1GB to force out-of-core computation even on PCs with large memory.

We demonstrate three complete reconstructions. Canton#1 and Canton#2, are shown in Figure 4. UNC sequence is shown in Figure 5. Some statistic is listed in Table 2. Typical input images are also shown in Figure 3. These

Table 1. Statistic on three moderate scale data sets. "spa pt.#" is the number of reconstructed sparse 3D points. "img.#" is the number of input images. "spa" is whether the sparse reconstruction successes or not. "qd pt. #" is the number of reconstructed semi-dense 3D points. "qd" is whether the semi-dense reconstruction successes or not. "md" is the minimal number of resampled points that still yields reasonable result visually using "Comp" score. The resample rate is included in the bracket. "size" is the size of input images.

seq	spa pt.#	img.#	spa	qd pt.#	qd	md	size
Hall	2325	113	Fail	206,094	Success	413 (0.2%)	1024×682
Building	115	10	Success	4,449	Success	36 (0.8%)	640×426
Oxford	653	11	Success	14,985	Success	49 (0.3%)	512×512

Table 2. Statistics on three large scale data sets. "seq." lists the names of 3 complete reconstructions. "img#" is the number of images used in the reconstruction. "tp.#" is the total number of points reconstructed. "rp.#" is the number of points that are used in final in-core computation. "rm" is the amount of memory used for the resampled reconstruction. "om" is the amount of memory that is needed to fit the bundle adjustment problem with all points (both in-core and out-of-core) and all cameras. "size" is the size of input images.

seq.	img.#	tp.#	rp.#	rm (GB)	om (GB)	size
Canton#1	344	6,420k	378k	0.63	10.1	2400×1600
Canton#2	277	3,819k	412k	0.62	5.56	2400×1600
UNC	921	5,639k	72k	0.4	15.2	1024×768

examples are reconstructed using the proposed out-of-core merging process. All intermediate merged results are bundled in the approximate manner we proposed. Only the bundle adjustment on the final results is carried out on all points and cameras. From the column "rm" and "om" in Table 2, we can see how our out-of-core merging process reduces the amount of memory used in bundle adjustment. Moreover, we make a comparison between a few different reconstruction methods on UNC in Figure 6. We further take the camera motion measured using GPS/INS system as a reference. The reconstruction of "SPA" and "Ran" both failed, while the camera motion reconstructed by "Comp" is very close to GPS/INS measurement even with fewer points. The failure occurs when two subsequences are merged because of the inconsistent reconstruction of the overlapping cameras of two subsequences.

4.4 Discussion

Global vs. local. The requirement of the density of the 3D points usually differs from application to application. For example, for image based modeling, it is better to reconstruct as many 3D points locally as possible to assist the modeling of each individual object. In contrast, the global reconstruction of camera poses is crucial not only for the registration of individual model into a global coordinate

system in large scale city modeling [21], but also for the application in localization and mapping. With our resampling framework, a global geometry computation can be first carried out with a lower density so that the very large-scale sequence can be handled, while local geometries can be densified again using the original detected matches and the estimated global geometry. This kind of level-of-detail relationship is illustrated in Figure 1 and the bottom rows in Figure 5 and Figure 4 . Because of our out-of-core merging process, all the points that cannot be put in memory are still on the hard disk. It is very easy to reuse these 3D points whenever they are needed.

Relation with other large scale methods. The results demonstrate that our resample method can scale up properly into very large-scale data set. However, as stated in other literature on large scale structure from motion [8,22], the running time and resource will be dominated by cameras when the number of cameras grows larger. Our work is complementary to the works targeting on reduced the redundancy in-between images.

5 Conclusion

We propose a hierarchical approach of mixing global and local geometries and controlling the on-demand density of 3D reconstruction. The mixture of global and local geometries is handled by the statistical analysis of the reconstruction accuracy and robustness from local to global. We studied our proposed resample scheme carefully through a few validation experiments. And our approach was also validated on the large-scale data set. The experiment results indicate that sampling with our score functions can obtain robust reconstruction similar to semi-dense approach, while the problem size is as neat as sparse approach. The trade off for this advantage is the extra computation time on match propagation and resampling compared to sparse approach.

Acknowledgments. This work was supported by the Hong Kong RGC GRF 618908 and RGC GRF 619409. We acknowledge Google for the images that were used in Figure 1, Vision Geometry Group, Oxford for the data set OXFORD, and University of North Carolina at Chapel Hill and University of Kentucky for the data set UNC. We also thank Maxime Lhuillier for helpful discussions.

References

1. Pollefeys, M., Nistér, D., Frahm, J., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S., Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewénius, H., Yang, R., Welch, G., Towles, H.: Detailed real-time urban 3D reconstruction from video. IJCV 78, 143–167 (2008)
2. Quan, L.: Invariants of six points and projective reconstruction from three uncalibrated images. IEEE PAMI 17, 34–46 (1995)
3. Nistér, D.: Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1842, pp. 649–663. Springer, Heidelberg (2000)

4. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3D. ACM Trans. Graph. 25, 835–846 (2006)
5. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment - a modern synthesis. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) ICCV-WS 1999. LNCS, vol. 1883, pp. 298–372. Springer, Heidelberg (2000)
6. Lhuillier, M., Quan, L.: A quasi-dense approach to surface reconstruction from uncalibrated images. IEEE PAMI 27, 418–433 (2005)
7. Nistér, D.: Frame decimation for structure and motion. In: Pollefeys, M., Van Gool, L., Zisserman, A., Fitzgibbon, A.W. (eds.) SMILE 2000. LNCS, vol. 2018, pp. 17–34. Springer, Heidelberg (2001)
8. Snavely, N., Seitz, S., Szeliski, R.: Skeletal graphs for efficient structure from motion. In: CVPR, pp. 1–8 (2008)
9. Ni, K., Steedly, D., Dellaert, F.: Out-of-core bundle adjustment for large-scale 3D reconstruction. In: ICCV, pp. 1–8 (2007)
10. Mouragnon, E., Dekeyser, F., Sayd, P., Lhuillier, M., Dhome, M.: Real time localization and 3D reconstruction. In: CVPR, vol. 1, pp. 363–370 (2006)
11. Eudes, A., Lhuillier, M.: Error propagations for local bundle adjustment. In: CVPR Workshops, pp. 2411–2418 (2009)
12. Seitz, S., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: CVPR, pp. 519–528 (2006)
13. Morris, D.D.: Gauge Freedoms and Uncertainty Modeling for Three-dimensional Computer Vision. PhD thesis, Carnegie Mellon University (2001)
14. Lowe, D.: Object recognition from local scale-invariant features. In: ICCV, pp. 1150–1157 (1999)
15. Levinshtein, A., Stere, A., Kutulakos, K., Fleet, D., Dickinson, S., Siddiqi, K.: TurboPixels: fast superpixels using geometric flows. IEEE PAMI 31, 2290–2297 (2009)
16. Baker, J.: Reducing bias and inefficiency in the selection algorithm. In: Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application Table of Contents, pp. 14–21. L. Erlbaum Assoiates Inc., Hillsdale (1987)
17. Lhuillier, M., Quan, L.: Robust dense matching using local and global geometric constraints. In: ICPR, pp. 968–972 (2000)
18. Lhuillier, M., Quan, L.: Match propagation for image-based modeling and rendering. IEEE PAMI 24, 1140–1146 (2002)
19. Nistér, D.: An efficient solution to the five-point relative pose problem. IEEE PAMI 26, 756–777 (2004)
20. Lourakis, M.A., Argyros, A.: SBA: A software package for generic sparse bundle adjustment. ACM Trans. Math. Software 36, 1–30 (2009)
21. Xiao, J., Fang, T., Zhao, P., Lhuillier, M., Quan, L.: Image-based street-side city modeling. ACM Trans. Graph. 28, 114:1–114:12 (2009)
22. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building rome in a day. In: ICCV, pp. 72–79 (2009)

Sequential Non-Rigid Structure-from-Motion with the 3D-Implicit Low-Rank Shape Model*

Marco Paladini[1], Adrien Bartoli[2], and Lourdes Agapito[1]

[1] Queen Mary University of London, Mile End Road, E1 4NS London, UK
[2] Clermont Université, France

Abstract. So far the Non-Rigid Structure-from-Motion problem has been tackled using a batch approach. All the frames are processed at once after the video acquisition takes place. In this paper we propose an incremental approach to the estimation of deformable models. Image frames are processed online in a sequential fashion. The shape is initialised to a rigid model from the first few frames. Subsequently, the problem is formulated as a model based camera tracking problem, where the pose of the camera and the mixing coefficients are updated every frame. New modes are added incrementally when the current model cannot model the current frame well enough. We define a criterion based on image reprojection error to decide whether or not the model must be updated after the arrival of a new frame. The new mode is estimated performing bundle adjustment on a window of frames. To represent the shape, we depart from the traditional explicit low-rank shape model and propose a variant that we call the 3D-implicit low-rank shape model. This alternative model results in a simpler formulation of the motion matrix and provides the ability to represent degenerate deformation modes. We illustrate our approach with experiments on motion capture sequences with ground truth 3D data and with real video sequences.

1 Introduction

The reconstruction of 3D scenes from monocular video sequences is one of the fundamental problems in computer vision. Following the success on rigid structure recovery in recent years there has been a wealth of research on modelling deformable structures. Most Non-Rigid Structure-from-Motion (NR SfM) algorithms to date rely on the foundational model proposed by Bregler *et al.* [4] which describes the time varying structure of a deforming object as a linear combination of basis shapes. The pose, the basis and the time varying coefficients are then estimated using a batch approach – all the frames in the sequence are processed at once after the acquisition.

While batch and real-time sequential rigid SfM are mature fields that have now consolidated into commercial applications, NR SfM is still at its infancy.

* This work was partially funded by the European Research Council under ERC Starting Grant agreement 204871-HUMANIS and the British Council/Alliance Research Programme. A. Bartoli was funded by ANR through the HFIBMR Project.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 15–28, 2010.

Some batch algorithms exist [3,13,10] but there is still a need to define deformable shape models and estimation algorithms that will allow to push NR SfM forward to a scenario where it might emulate the successes of its rigid counterpart, in terms of robust performance and application to real world cases. In this paper we advance the state of the art in NR SfM in two main directions, both proposing a new sequential estimation paradigm and an alternative low-rank shape model.

Our first contribution is the definition of a new estimation paradigm that extends NR SfM to the sequential domain. We propose a rank-growing engine which will determine when the rank of the model should be increased and if necessary will estimate the new mode.

We divide the sequential non-rigid shape estimation into two processes: model-based tracking of the camera pose and shape coefficients and model update. The first process assumes that a current up-to-date model, of a certain rank, of the 3D shape observed so far exists and performs *model based camera tracking*: when a new frame arrives this module estimates the current camera pose and the shape parameters using as input the 2D coordinates of image features matched in the last W frames, where W is the width of a sliding window. The second process is a *model update* module which decides, based on the image reprojection error given by the camera tracking module, whether or not the current model is able to explain the deformations viewed in the new frame. If the current model does not have enough descriptive power to capture the deformations observed in the new frame, the model update module will add a new mode and estimate its parameters using bundle adjustment on a sliding window. The entire system is bootstrapped from a rigid reconstruction obtained from a small number of initial frames.

Our second contribution is an alternative low-rank shape model that provides the ability to represent modes of deformation of dimensionality lower than 3 (for instance deformations on a plane or along a line).

We call it the *3D implicit low-rank shape model* since it does not use an explicitly defined 3D shape basis. This has two main advantages. First, the motion matrix in our model has a simpler structure than in the classical model, which allows for a linear estimation of camera pose and shape coefficients from a single frame, and can be used to initialise the bundle adjustment in the sequential framework. Second, our model handles deformations whose rank is not a multiple of 3 and thus avoids one to explicitly compute the rank of a particular shape basis. When the deformations are processed one frame at a time, having the flexibility to update the model with 1-dimensional modes fits the sequential estimation paradigm more naturally, since there is a much higher chance of observing lower dimensional deformations.

It is important to note that in this paper we do not try to solve the matching problem. Instead, we rely on point correspondences between frames being available. The integration of the feature tracking problem with the camera tracking and model update processes (which are the focus of this paper) is beyond the scope of this work although we certainly intend to address it in our future work.

2 Related Work

The ability to reconstruct a deformable 3D surface from a monocular sequence when the only input information is a set of point correspondences between images is an ill posed problem unless more constraints than just the reprojection error are used. The seminal work of Bregler *et al.* [4] was the first to propose a solution to the NR SfM problem for the orthographic camera case. This model not only provided an elegant extension of the rigid factorisation framework [12] but has also opened up new computational and theoretical challenges in the field.

Current solutions to NR SfM focus on the definition of optimization criteria to guarantee the convergence to a well behaved solution. This is often only achieved through the addition of temporal and spatial smoothness priors. Bundle adjustment has become a popular optimization tool to refine an initial rigid solution while incorporating temporal and spatial smoothness priors on the motion and the deformations.

Aanaes *et al.* [1] were the first to formulate the problem using bundle adjustment using smoothness priors. Later, Del Bue *et al.* [5] incorporated the constraint that some of the points on the object were rigid while Bartoli *et al.* [3] used a coarse to fine shape model where new deformation modes are added iteratively to capture as much of the variance left unexplained by previous modes as possible. Torresani *et al.* [13] formulate the problem using Probabilistic Principal Components Analysis introducing priors as a Gaussian distribution on the deformation weights. More recently, Paladini *et al.*'s [10] work focuses on ensuring that the solution lies on the correct motion manifold where the metric constraints are exactly satisfied. All these approaches are initialised from a rigid solution and they use temporal and spatial smoothness priors on the motion and shape parameters. Olsen *et al.* [9] proposed the surface shape prior and an implicit model that simplifies the estimation process but leads to a non-Euclidean 3D reconstruction.

The linear subspace model has also allowed closed-form solutions to be proposed for the cases of both affine [14] and perspective [16,6] cameras. Recently, a set of new approaches has departed from the low-rank linear shape model. Rabaud and Belongie [11] adopt a manifold learning framework assuming that only small neighbourhoods of shapes are well modelled with a linear subspace.

Akhter *et al.* [2] described the structure of a non-rigid body in trajectory space as a linear combination of DCT basis trajectories with the obvious advantage that the basis is object independent.

The common attribute to all NR SfM algorithms proposed so far is that they are batch methods. Our new sequential approach is motivated by recent developments in the area of sequential real-time SfM methods for rigid scenes [7,8]. In particular, our approach is inspired by the work of Klein and Murray [7] in which they develop a real time system based on two parallel threads – the camera tracking thread which performs real time model based pose estimation and the mapping thread which runs in a constant loop performing bundle adjustment on a small set of key-frames. To the best of our knowledge our work is the first in NR SfM to depart from the batch formulation and reformulate the shape

estimation sequentially. First we introduce a new variant to the low-rank linear basis shape model that we believe is better suited to a sequential formulation.

3 New Deformation Model

3.1 Classical Explicit Low-Rank Shape Model

In the case of deformable objects the observed 3D points change as a function of time. In the low-rank shape model defined by Bregler *et al.* [4] the 3D points deform as a linear combination of a fixed set of K rigid shape bases according to time varying coefficients. In this way, $\mathtt{S}_f = \sum_{k=1}^{K} l_{fk}\mathtt{B}_k$ where the matrix $\mathtt{S}_f = [\mathbf{X}_{f1}, \cdots \mathbf{X}_{fP}]$ contains the 3D coordinates of the P points at frame f, the $3 \times P$ matrices $\mathtt{B}_k$ are the shape bases and l_{fk} are the coefficient weights. If the 3D shape is known, this model can be obtained from the PCA decomposition of the $\mathtt{S}^*$ that contains the 3D shape in all the frames.

$$\mathtt{S}^*_{F\times 3P} = \begin{bmatrix} \mathtt{S}^*_1 \\ \mathtt{S}^*_2 \\ \vdots \\ \mathtt{S}^*_F \end{bmatrix} = \begin{bmatrix} X_{11} & Y_{11} & Z_{11} & \cdots & X_{1P} & Y_{1P} & Z_{1P} \\ & \vdots & & & & \vdots & \\ X_{F1} & Y_{F1} & Z_{F1} & \cdots & X_{FP} & Y_{FP} & Z_{FP} \end{bmatrix} \tag{1}$$

A PCA decomposition of rank K of $\mathtt{S}^*$ would give $\mathtt{LB}^*$, where $\mathtt{L}$ is the $F \times K$ matrix of deformation weights l_{ik}, and the $K \times 3P$ matrix $\mathtt{B}^*$ can be rearranged to give the basis shapes $\mathtt{B}_k$. If we assume an orthographic projection model the coordinates of the 2D image points observed at each frame i are then given by:

$$\mathtt{W}_i = \mathtt{R}_i \left(\sum_{k=1}^{K} l_{ik}\mathtt{B}_k \right) + \mathtt{T}_i \tag{2}$$

where $\mathtt{R}_i$ is a 2×3 *Stiefel matrix* and $\mathtt{T}_i$ aligns the image coordinates to the image centroid. The aligning matrix $\mathtt{T}_i$ is such that $\mathtt{T}_i = \mathbf{t}_i \mathbf{1}_P^T$ where the 2-vector $\mathbf{t}_i$ is the 2D image centroid and $\mathbf{1}_P$ a vector of ones.

When the image coordinates are registered to the centroid of the object and we consider all the frames in the sequence, we may write the measurement matrix as:

$$\mathtt{W} = \begin{bmatrix} l_{11}\mathtt{R}_1 & \dots & l_{1K}\mathtt{R}_1 \\ \vdots & \ddots & \vdots \\ l_{F1}\mathtt{R}_F & \dots & l_{FK}\mathtt{R}_F \end{bmatrix} \begin{bmatrix} \mathtt{B}_1 \\ \vdots \\ \mathtt{B}_K \end{bmatrix} = \mathtt{MS} \tag{3}$$

Since $\mathtt{M}$ is a $2F \times 3K$ matrix and $\mathtt{S}$ is a $3K \times P$ matrix in the case of deformable structure the rank of $\mathtt{W}$ is constrained to be at most $3K$. The motion matrices now have a complicated repetitive structure $\mathtt{M}_i = [\mathtt{M}_{i1} \dots \mathtt{M}_{iK}] = [l_{i1}\mathtt{R}_i \dots l_{iK}\mathtt{R}_i]$ that makes the model estimation difficult.

Olsen *et al.* [9] proposed to consider an implicit model where the repetitive structure of the motion matrix is not used. While this simplifies the estimation problem, the recovered model does not directly provide usable motion and shape parameters, unless a mixing matrix is computed [4,14]. The mixing matrix computation problem has not received a simple solution so far.

3.2 Proposed 3D-Implicit Low-Rank Shape Model

In this paper we propose to depart from the traditional basis shapes model, and embrace a different formulation that will fit the problem of sequential structure recovery more naturally since it allows for the rank of the shape model to grow one by one with the arrival of a new frame, instead of multiples of three.

The data in the shape matrix may be re-arranged in a different form, stacking the shape matrices vertically for all frames F. Each matrix $\mathtt{S}_f \in \mathbb{R}^{3\times P}$ contains the 3D coordinates of P points in frame f.

$$\mathtt{S}_{3F\times P} = \begin{bmatrix} \mathtt{S}_1 \\ \mathtt{S}_2 \\ \vdots \\ \mathtt{S}_F \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} & & X_{1P} \\ Y_{11} & Y_{12} & \cdots & Y_{1P} \\ Z_{11} & Z_{12} & & Z_{1P} \\ \vdots & \vdots & & \vdots \\ X_{F1} & X_{F2} & & X_{FP} \\ Y_{F1} & Y_{F2} & \cdots & Y_{FP} \\ Z_{F1} & Z_{F2} & & Z_{FP} \end{bmatrix} \tag{4}$$

If we assume that the shape matrix $\mathtt{S}$ is low-rank we can perform Principal Components Analysis to obtain a PCA basis as $\mathtt{S} = \mathtt{U}_d\mathtt{V}_d$, where d is the rank of the decomposition, $\mathtt{U}_d \in \mathbb{R}^{3F\times d}$ and $\mathtt{V}_d \in \mathbb{R}^{d\times P}$. We can also explicitly include an average rigid (mean) shape in the model, therefore the shape at frame f would be given by:

$$\mathtt{S}_f = \bar{\mathtt{S}} + \begin{bmatrix} \mathbf{U}_{f1} \cdots \mathbf{U}_{fr} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \vdots \\ \mathbf{V}_r \end{bmatrix} \tag{5}$$

where $\bar{\mathtt{S}}$ is the mean shape, $d = 3+r$, $\mathbf{U}_{fr}$ is the 3-vector $[U(x)_{fr}U(y)_{fr}U(z)_{fr}]^T$ and $\mathbf{V}_r$ are the rows of matrix $\mathtt{V}$.

Therefore we can consider $\mathtt{V}$ to be a PCA basis of the shape (row) space of $\mathtt{S}$ and $\mathtt{U}$ to contain the time varying coefficients. Note that in this case the shape matrix $\mathtt{V}$ has dimensions $r \times P$ where r is the rank of the decomposition and P is the number of points in the shape. For each frame $3r$ coefficients are needed to express the configuration of the shape.

We assume that the shape at instant f is then projected onto an image following an orthographic camera model. The 2D coordinates of the points can then be expressed as:

$$\mathtt{W}_f = \begin{bmatrix} u_{f1} \cdots u_{fP} \\ v_{f1} \cdots v_{fP} \end{bmatrix} = \mathtt{R}_f\mathtt{S}_f + \mathbf{T}_f = \mathtt{R}_f(\bar{\mathtt{S}} + \mathtt{U}_f\mathtt{V}) + \mathbf{T}_f \tag{6}$$

where $\mathtt{R}_f$ is a $[2 \times 3]$ orthographic camera projection matrix, it encodes the first two rows of the camera rotation matrix and $\mathtt{T}_f$ the translation for frame f. If we

now register all the measurements to their centroid in each frame the projection of the shape in all frames can be written as:

$$\mathtt{W} = \begin{bmatrix} \mathtt{R}_1 & & \\ & \mathtt{R}_2 & \\ & & \ddots \\ & & & \mathtt{R}_F \end{bmatrix} \left(\begin{bmatrix} \bar{\mathtt{S}} \\ \bar{\mathtt{S}} \\ \vdots \\ \bar{\mathtt{S}} \end{bmatrix} + \begin{bmatrix} \mathbf{U}_{11} & \cdots & \mathbf{U}_{1r} \\ \mathbf{U}_{21} & \cdots & \mathbf{U}_{2r} \\ \vdots & & \vdots \\ \mathbf{U}_{F1} & \cdots & \mathbf{U}_{Fr} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \vdots \\ \mathbf{V}_r \end{bmatrix} \right) \tag{7}$$

In our model, the basis shapes are not explicitly used as in the classical model, while the camera projection is explicitly modeled. We thus call our model the *3D-implicit low-rank shape model*. Our model combines Bregler *et al.* [4]'s explicit model and Olsen *et al.* [9]'s implicit model. It has the following two main advantages:

1. **Simplicity.** The motion matrix is block diagonal and only contains the rotation matrices instead of a mixture of the coefficients and the rotations. The fact that the 3D basis is not explicitly available in our model is not a problem since one is generally more interested in recovering the 3D shape of the observed scene than the basis shapes – the basis shapes can be estimated a posteriori by forming and factorizing the matrix $\mathtt{S}^*$ in equation (1). As we explain below, it also is an advantage not to have explicit 3D basis shapes.
2. **Any-rank deformations.** Our formulation allows us to define shape models where the rank is not a multiple of 3. In other words, in the explicit model, a basis shape always has to be of rank 3, whereas in the real world not all deformations are of rank 3. Xiao and Kanade [15] propose to explicitly find the rank of a particular deformation mode (which can be one of 1, 2 or 3). Our model circumvents this difficult problem.

4 A Sequential Approach to NR SfM

In this paper we depart from the batch formulation of NR SfM and we propose a sequential approach based on the alternative low-rank shape model outlined in the previous section. Our approach can be seen as a two process formulation. The system holds a current up-to-date model, of a certain rank, encapsulated in matrix $\mathtt{V}$. The first process is a model based camera tracking module. Given the current estimate of $\mathtt{V}$, when a new frame arrives, the camera tracking module estimates the new pose $\mathtt{R}_f$ and the new deformation coefficients $\mathtt{U}_f$ for the current frame. If the current model explains well the measurements the image reprojection error will be low. However, if the error goes above some defined threshold the rank of the model must be increased and the model updated. I that case, a model update module will update the current model adding a new row to matrix $\mathtt{V}$. As the sequence is processed the model will become more complicated, until all the possible object deformations have been observed. Our sequential approach to NR SfM is summarised in Algorithm 1. We now describe in detail the two main modules of our sequential system: the camera tracking module and the model update module.

Algorithm 1. Sequential Non-Rigid Structure-from-Motion (NR SfM)

Input: 2D point correspondences
Output: 3D coordinates of the deforming surface for each frame.

```
Initialise model to mean rigid shape S̄ estimated via rigid factorization on the first few frames.
loop
    new frame f arrives
    run camera tracking process: estimate camera pose R_i and coefficients U_i
    while (image reprojection error is above threshold) do
        run model update process:
            increase rank r ← r + 1
            estimate new row of V and new column of U_f
    end while
    go to process next frame; f ← f + 1
end loop
```

5 Camera Tracking Given a Known Model V

If the matrix $\mathtt{V}$ is known in advance, the NR SfM problem is reduced to the estimation of the camera pose $\mathtt{R}_f$ and the mixing coefficients $\mathtt{U}_f$ for each frame. In that case, the pose of the camera and the coefficients can be updated sequentially for each frame using a model based approach.

We adopt a sliding window approach where we perform bundle adjustment on the last W frames where W is the width of a pre-defined window. The cost to be minimised is the image reprojection error over all frames in the window:

$$\min_{\mathtt{R}_i,\mathtt{U}_i} \sum_{i=f-W}^{f} \|\mathtt{W}_i - \mathtt{R}_i(\bar{\mathtt{S}} + \mathtt{U}_i\mathtt{V})\|_F^2 \tag{8}$$

To this cost function we add a temporal smoothness prior to penalise strong variations in the camera matrices of the form $\|\mathtt{R}_i - \mathtt{R}_{i-1}\|_F^2$, and a shape smoothness prior (similar to the one used in [3]) that ensures that points that lie close to each other in space should stay close. The shape smoothness is defined as $\sum_{i=f-W}^{f} D^{i,i-1}$, where $D^{i,i-1}$ is the change in the euclidean distance between 3D points over two frames: $D^{i,i-1} = \sum_{a,b=1}^{P} \phi_{a,b} |d^2(\mathbf{X}_{i,a}, \mathbf{X}_{i,b}) - d^2(\mathbf{X}_{i-1,a}, \mathbf{X}_{i-1,b})|$. The weight $\phi_{a,b}$ is a measure of the closeness of points a and b, defined as a $P \times P$ affinity matrix $\phi_{a,b} = \rho(d^2(\mathbf{X}_a, \mathbf{X}_b))$ where ρ is a truncated Gaussian kernel. The final cost function can now be written as:

$$\min_{\mathtt{R}_i,\mathtt{U}_i} \sum_{i=f-W}^{f} \|\mathtt{W}_i - \mathtt{R}_i(\bar{\mathtt{S}} + \mathtt{U}_i\mathtt{V})\|_F^2 + \lambda \sum_{i=f-W}^{f} \|\mathtt{R}_i - \mathtt{R}_{i-1}\|_F^2 + \psi \sum_{i=f-W}^{f} D^{i,i-1} \tag{9}$$

The mean shape $\bar{\mathtt{S}}$ and the shape model $\mathtt{V}$ are assumed to be known. This non-linear minimization requires an initial estimate for the camera pose $\mathtt{R}_f$ and the shape coefficients $\mathtt{U}_f$ in the current frame f. Algorithms to obtain linear estimates for $\mathtt{R}_f$ and $\mathtt{U}_f$ are described in Section 5.1.

The steps of the complete algorithm to track the current pose of the camera and the shape coefficients given the shape model can be summarised as follows. Each time a new frame f of feature tracks is available:

- Obtain initial estimates for the current pose $\mathtt{R}_f$ and mixing coefficients $\mathtt{U}_f$ using the linear estimation plus prior described in Section 5.1.
- Minimize the cost function (9) with smoothness priors using bundle adjustment to obtain optimized values for the rotations $\mathtt{R}_i$ and shape coefficients $\mathtt{U}_i$ in all the frames in the sliding window.
- If the reprojection error of the window becomes higher than a threshold, signal the modelling process to increase the rank of the $\mathtt{V}$ matrix.

5.1 Initialization: Linear Estimation of $\mathtt{U}_f$ and $\mathtt{R}_f$

Consider new image measurements become available for a new frame. These can be arranged in a $2 \times P$ matrix for that single frame called $\mathtt{W}_f$. The projection model gives us the relation $\mathtt{W}_f = \mathtt{R}_f(\bar{\mathtt{S}} + \mathtt{U}_f\mathtt{V}) + \mathtt{T}_f$.

Linear estimation of $\mathtt{R}_f$. For every new frame the camera pose $\mathtt{R}_f$ must be initialised before Bundle Adjustment. For this purpose, we approximate the shape with the rigid mode to obtain an initial estimate of the camera rotation. This means we need to find the camera pose $\mathtt{R}_f$ that satisfies $\mathtt{W}_f = \mathtt{R}_f\mathtt{S}$, while respecting the smoothness prior $\lambda\mathtt{I}\,\mathrm{vec}(\mathtt{R}_f) = \lambda\,\mathrm{vec}(\mathtt{R}_{f-1})$. Using the relation $\mathrm{vec}(\mathtt{AXB}) = [\mathtt{B}^T \otimes \mathtt{A}]\,\mathrm{vec}(\mathtt{X})$, where $\otimes$ is the Kronecker product and vec(.) is the column-major vectorisation of a matrix, and using $\mathtt{W}_f = \mathtt{I}_2\mathtt{R}_f\mathtt{S}$ we can write:

$$\mathrm{vec}(\mathtt{W}_f) = [\mathtt{S}^T \otimes \mathtt{I}_2]\,\mathrm{vec}(\mathtt{R}_f) \tag{10}$$

$$\begin{bmatrix} [\mathtt{S}^T \otimes \mathtt{I}_2] \\ \lambda\mathtt{I} \end{bmatrix} \mathrm{vec}(\mathtt{R}_f) = \begin{bmatrix} \mathrm{vec}(\mathtt{W}_f) \\ \lambda\,\mathrm{vec}(\mathtt{R}_{f-1}) \end{bmatrix} \tag{11}$$

The resulting $\mathtt{R}_f$ will not be orthonormal (i.e. not a truncated rotation matrix), so we find the closest orthonormal rigid projection using SVD.

Linear estimation of $\mathtt{U}_f$. First we take away the contribution to the image measurements given by the known translation and mean shape component to give $\tilde{\mathtt{W}}_f = \mathtt{W}_f - \mathtt{T}_f - \mathtt{R}_f\bar{\mathtt{S}} = \mathtt{R}_f\mathtt{U}_f\mathtt{V}$, which can be rewritten as $\mathrm{vec}(\tilde{\mathtt{W}}_f) = [\mathtt{V}^T \otimes \mathtt{R}_f]\,\mathrm{vec}(\mathtt{U}_f)$. This provides a linear equation on the unknown vector $\mathtt{U}_f$. However, this is not sufficient to produce an acceptable solution, because $\mathtt{U}_f$ is a $3 \times r$ matrix where each column $\mathbf{U}_{fr}$ is a 3-vector $[U(x)_{fr}U(y)_{fr}U(z)_{fr}]^T$ that contains the PCA coefficients of all 3D coordinates, while $\tilde{\mathtt{W}}_f$ contains 2D projections. However, this problem can be overcome by including a temporal smoothness prior term that penalises solutions that are far from the value for the previous frame $\mathtt{U}_{f-1}$. Thus the prior term is of the form $\lambda\mathtt{I}\,\mathrm{vec}(\mathtt{U}_f) = \lambda\,\mathrm{vec}(\mathtt{U}_{f-1})$. We can join both linear equations and solve the linear system:

$$\begin{bmatrix} [\mathtt{V}^T \otimes \mathtt{R}_f] \\ \lambda\mathtt{I} \end{bmatrix} \mathrm{vec}(\mathtt{U}_f) = \begin{bmatrix} \mathrm{vec}(\tilde{\mathtt{W}}_f) \\ \lambda\,\mathrm{vec}(\mathtt{U}_{f-1}) \end{bmatrix} \tag{12}$$

6 Sequential Update of the Shape Model

In NR SfM the 3D object the camera observes varies over time. The current model will encode the modes of deformation that the object has exhibited so far in the sequence. However, if the object deforms in different ways that are not encoded in the model the camera tracking will fail. Therefore, a mechanism is needed to update the model when new modes of deformation appear. In that case, the rank of the model should grow and the parameters of the model should be fit to the new data.

The difficulty of updating the model in an sequential way is doublefold. Firstly, when each new frame arrives, we need a mechanism to decide whether or not the current model continues to fit the data well enough. While the shape model can still describe the data, we can continue to do model based camera tracking. We decide this based on the image reprojection error. Secondly, if the model can no longer explain the data, the rank of the model needs to grow to incorporate the new mode of deformation and the parameters of the new row of $\mathtt{V}$ and the new column of $\mathtt{U}$ must be estimated.

6.1 Rank Increase Criterion

The rank selection criterion will decide to increase the rank only if the current data does not fit the model well enough, i.e. if the existing modes do not model the current frame well. Therefore we use the image reprojection error as the criterion – if the error increases above a certain threshold we increase the rank of the shape model. This results in a new row being added to the PCA basis $\mathtt{V}$ and a new column to the PCA components $\mathtt{U}$. However, the new mode is recovered from the current frame only, so it has no influence over past frames. Therefore for all past frames we can set the $3(f-1)$ components of the new column of $\mathtt{U}$ to 0.

6.2 Model Update: Estimating New Row of $\mathtt{V}$ and New Column of $\mathtt{U}$

When the camera tracking module processes a new frame that it cannot model well enough (the reprojection error is above the defined threshold), the model is updated by increasing the rank. Ideally once all the different modes of deformation that an object can exercise are incorporated in the PCA basis, the rank will remain stable and the camera tracking process will be able to reconstruct the incoming frames.

Given new image correspondences for frame f, the rank of $\mathtt{U}$,$\mathtt{V}$ must be increased. From the current estimate of $\mathtt{U}_{f,1:r-1}$ and $\mathtt{V}_{1:r-1}$ we can rewrite the model for the new frame as

$$\tilde{\mathtt{W}}_f = \mathtt{R}_f(\bar{\mathtt{S}} + \mathtt{U}_{f,1:r-1}\mathtt{V}_{1:r-1} + \mathtt{U}_{f,r}\mathtt{V}_r). \tag{13}$$

Both the residual of the current model $\mathtt{A} = \tilde{\mathtt{W}}_f - \mathtt{R}_f\,(\bar{\mathtt{S}} + \mathtt{U}_{f,1:r-1}\mathtt{V}_{1:r-1})$ and the current camera rotation $\mathtt{R}_f$ are known. We need to estimate $\mathtt{Z} = \mathtt{U}_{f,r}\mathtt{V}_r$, the contribution of the new rank, subject to the following constraints:

$$\mathtt{A} = \mathtt{R}_f\mathtt{Z} \quad \text{rank}(\mathtt{Z}) = 1 \tag{14}$$

This problem is difficult to solve in closed form, therefore we approximate it using a linear solution as follows. We define $\mathtt{C}$ as the closest rank-1 approximation of $\mathtt{A}$ obtained using SVD, then compute $\mathtt{Z}$ as $\mathtt{Z} = \mathtt{R}_f^{\dagger}\mathtt{C}$. Finally, we can decompose $\mathtt{Z}$ using a rank-1 SVD decomposition to obtain a new row for $\mathtt{V}$.

Non-linear refinement. Once initial estimates are available for the new row of $\mathtt{V}$ and the new column of $\mathtt{U}$, they can be refined minimising image reprojection error over a sliding window of W frames

$$\min_{\mathtt{V}_r, \mathtt{U}_{ir}} \sum_{i=f-W}^{f} \|\mathtt{W}_i - \mathtt{R}_i(\bar{\mathtt{S}} + \mathtt{U}_i\mathtt{V})\|_F^2 \tag{15}$$

incorporating the smoothness priors described in section 5. Once the model is updated, the camera tracking module can resume *model based tracking* with the new model $\mathtt{V}$ with rank $r + 1$.

6.3 Bootstrapping

One of the known challenges in sequential approaches to rigid SfM is the initialization [7]. It is common to run the system in batch mode for a few frames to obtain a first model of the scene before starting the sequential operation. In the current experiments we run a rigid factorization algorithm on a few initial frames to obtain the rigid mean shape $\bar{\mathtt{S}}$. Once this is available the camera tracking and model update loop can start. An alternative approach that does not require manual intervention is the following. Start performing rigid factorization in batch. When a new frame arrives, if the reprojection error of rigid factorization over the frames observed so far is below the threshold then we keep performing rigid factorization. However, if the error becomes higher than our threshold, the mean shape of the non-rigid model is set to the rigid model obtained so far and we start our sequential NR SfM algorithm.

7 Experiments

7.1 Motion Capture Sequence *CMU-Face*

First we tested our sequential method based on the 3D-implicit low-rank shape model on a motion capture sequence with ground truth data[1]. This sequence from the CMU Motion Capture Database[2] contains 316 frames of motion capture data of the face of a subject wearing 40 markers performing deformations while rotating. This sequence was also used by Torresani *et al.* [13] to perform quantitative tests with ground truth data. We projected the 3D data synthetically using an orthographic camera model.

[1] Videos of the experimental results can be found on the project website `http://www.eecs.qmul.ac.uk/~lourdes/SequentialNRSFM`

[2] Available from `http://mocap.cs.cmu.edu`

Prior to the start of our sequential algorithm and with the purpose of bootstrapping the camera tracking module, we ran a batch rigid SfM algorithm [12] on the first 60 frames of the sequence to estimate the mean shape $\bar{\mathbf{S}}$. The PCA basis matrix $\mathtt{V}$ was initialised to $\mathtt{0}$. We then ran our new sequential algorithm based on the camera tracking and the model update modules, together with the rank detection engine. The average 3D error is 2.9%, with a 0.7 pixels 2D reprojection error on the 600×600pixels images. The reprojection threshold was fixed to 1.2pixels.

In Figure 1 we show results of the rank estimation, the 2D image reprojection error and the 3D error for each frame in the sequence using our sequential estimation formulation. The average image reprojection error over the whole sequence is less than a pixel. In Figure 3 (left) we compare results of the 3D error obtained with our method (Sequential), with Torresani *et al.*'s state of the art batch NR SfM algorithm (EM-LDS) [13].We show the histogram of 3D error

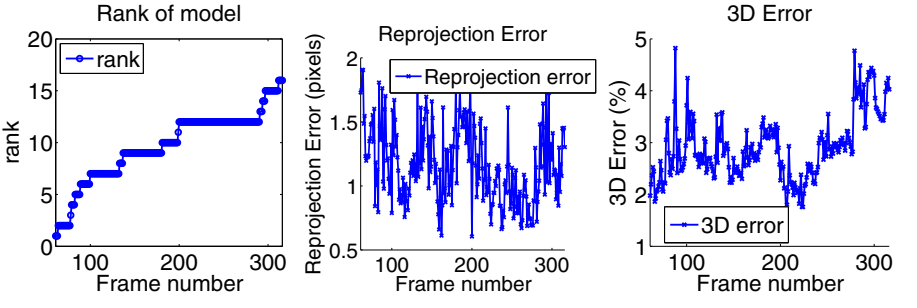


Fig. 1. Results of sequential NR SfM on the CMU-face sequence. Left: Value of the rank of the model for each frame, increasing as more frames are processed. Middle: 2D Reprojection error given by the camera tracking process. Right: 3D error of the reconstruction for each frame.

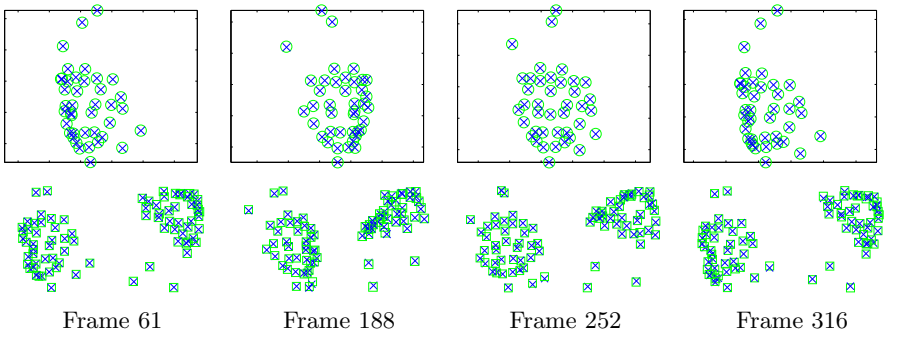


Fig. 2. 3D Reconstruction results obtained on the *CMU-face* sequence using camera tracking and model updating. First row: 2D image points (green circles) and reprojections (blue crosses). Second row: Views of the 3D reconstruction (crosses) compared with ground truth MOCAP data (squares).

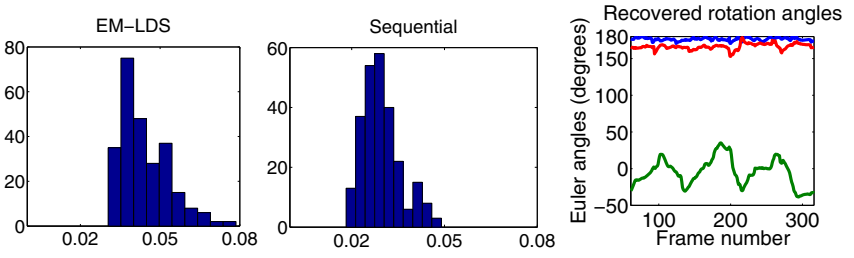


Fig. 3. (Left) Histogram of 3D error values built from all the frames, comparing results of our method (Sequential) with Torresani *et al.*'s state of the art batch (EM-LDS) [13]. The 3D errors obtained with our Sequential approach are comparable to the results from the batch method EM-LDS. (Right) Rotation angles estimated with the camera tracking module.

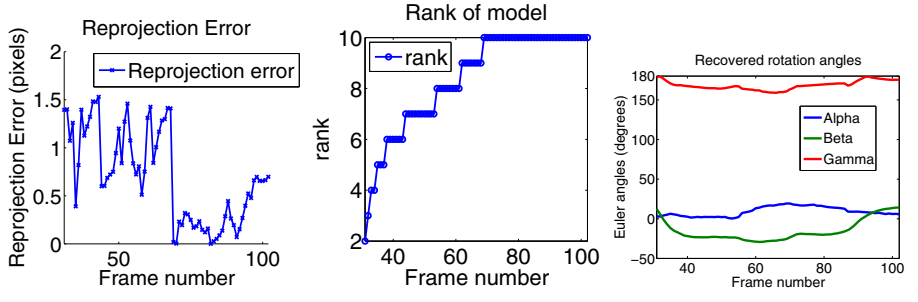


Fig. 4. Results on the *actress* sequence. Left: Reprojection error of the frame-by-frame reconstruction obtained with our method. Middle: The value of the rank, increased as more frames are processed. Right: Rotation angles estimated with the camera tracking module.

values taking into account all the frames in the sequence. The results show that our new sequential algorithm provides results comparable to Torresani *et al.*'s [13] batch state of the art algorithm. We show smooth estimates of the rotation angles for all the frames in the sequence in Figure 3 (right). In Figure 2 we show the 2D image reprojection error and the 3D reconstructions (blue crosses) we obtained for some frames in the sequence comparing them with ground truth values (green squares).

7.2 Real Data

We used the *actress* sequence, also used by Bartoli *et al.* [3], which consists of 102 frames of a video showing an actress talking and moving her head. In Figure 5 we show results of the 3D reconstructions obtained for some of the frames in the sequence. The camera tracking was bootstrapped with a rigid model obtained using Tomasi and Kanade's rigid factorization algorithm [12] on the first 30

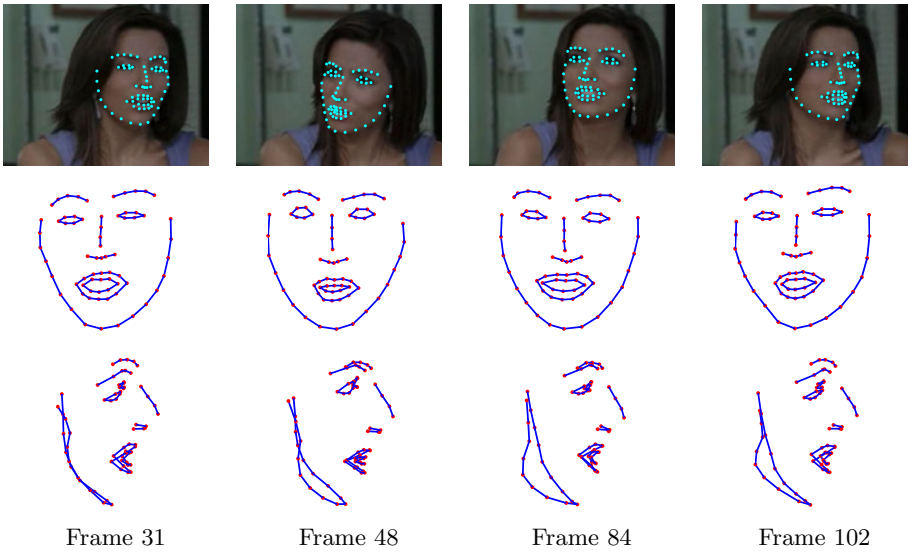


Fig. 5. Qualitative results on the *actress* sequence using camera tracking and model update. First row: The input images with superimposed feature tracking data. Second and Third rows: Front and side views of the 3D reconstruction of 4 frames of the sequence.

frames. The threshold for increasing the rank was a reprojection error of 0.9 pixels. From figure 4 we can see that the rank is increased, and the estimation of new frame parameters keeps the reprojection error low.

8 Conclusions

We have undergone a re-thinking of the NR SfM problem for monocular sequences providing a sequential solution. Our new sequential algorithm is able to automatically detect and increase the complexity of the model. Current state of the art methods for NR SfM are batch and rely on prior knowledge of the model complexity (usually the number of basis shapes, K). Our 3D-implicit low-rank shape model simplifies the projection model and allows the rank to grow one-by-one making it well suited to frame-by-frame operation. We have shown quantitative results on a motion capture sequence and shown our system in operation on a real sequence. Future work will pursue the goal of merging the feature tracking and modelling of image data into a single process. Concerning real time capability, our current MATLAB implementation is not real time. However, the sliding window approach ensures that the computation time per frame is bounded i.e. it does not grow with the number of frames. Therefore we foresee that with appropriate code optimisation we would be able to achieve real-time performance.

References

1. Aanæs, H., Kahl, F.: Estimation of deformable structure and motion. In: Workshop on Vision and Modelling of Dynamic Scenes, Copenhagen, Denmark (2002)
2. Akhter, I., Sheikh, Y., Khan, S., Kanade, T.: Nonrigid Structure from Motion in Trajectory Space. In: Neural Information Processing Systems (2008)
3. Bartoli, A., Gay-Bellile, V., Castellani, U., Peyras, J., Olsen, S., Sayd, P.: Coarse-to-Fine Low-Rank Structure-from-Motion. In: IEEE Conf. on Computer Vision and Pattern Recognition, Anchorage, Alaska (2008)
4. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3D shape from image streams. In: IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head, South Carolina (2000)
5. Del Bue, A., Lladó, X., Agapito, L.: Non-rigid metric shape and motion recovery from uncalibrated images using priors. In: IEEE Conf. on Computer Vision and Pattern Recognition, New York, NY (2006)
6. Hartley, R., Vidal, R.: Perspective nonrigid shape and motion recovery. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 276–289. Springer, Heidelberg (2008)
7. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2007), Nara, Japan (November 2007)
8. Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., Sayd, P.: Generic and real-time structure from motion using local bundle adjustment. Image Vision Comput. 27(8), 1178–1193 (2009)
9. Olsen, S.I., Bartoli, A.: Implicit non-rigid structure-from-motion with priors. J. Math. Imaging Vis. 31(2-3), 233–244 (2008)
10. Paladini, M., Del Bue, A., Stosic, M., Dodig, M., Xavier, J., Agapito, L.: Factorization for non-rigid and articulated structure using metric projections. In: IEEE Conf. on Computer Vision and Pattern Recognition, Miami, Florida (2009)
11. Rabaud, V., Belongie, S.: Re-thinking non-rigid structure from motion. In: IEEE Conf. on Computer Vision and Pattern Recognition, Anchorage, Alaska (2008)
12. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: A factorization approach. International Journal of Computer Vision 9(2) (1992)
13. Torresani, L., Hertzmann, A., Bregler, C.: Non-rigid structure-from-motion: Estimating shape and motion with hierarchical priors. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(5) (2008)
14. Xiao, J., Chai, J., Kanade, T.: A closed-form solution to non-rigid shape and motion recovery. International Journal of Computer Vision 67(2) (2006)
15. Xiao, J., Kanade, T.: Non-rigid shape and motion recovery: Degenerate deformations. In: IEEE Conf. on Computer Vision and Pattern Recognition, Washington D.C (2004)
16. Xiao, J., Kanade, T.: Uncalibrated perspective reconstruction of deformable structures. In: 10th Int. Conf. on Computer Vision, Beijing, China (2005)

Bundle Adjustment in the Large

Sameer Agarwal[1,⋆], Noah Snavely[2], Steven M. Seitz[3], and Richard Szeliski[4]

[1] Google Inc.
[2] Cornell University
[3] Google Inc. & University of Washington
[4] Microsoft Research

Abstract. We present the design and implementation of a new inexact Newton type algorithm for solving large-scale bundle adjustment problems with tens of thousands of images. We explore the use of Conjugate Gradients for calculating the Newton step and its performance as a function of some simple and computationally efficient preconditioners. We show that the common Schur complement trick is not limited to factorization-based methods and that it can be interpreted as a form of preconditioning. Using photos from a street-side dataset and several community photo collections, we generate a variety of bundle adjustment problems and use them to evaluate the performance of six different bundle adjustment algorithms. Our experiments show that truncated Newton methods, when paired with relatively simple preconditioners, offer state of the art performance for large-scale bundle adjustment. The code, test problems and detailed performance data are available at `http://grail.cs.washington.edu/projects/bal`.

Keywords: Structure from Motion, Bundle Adjustment, Preconditioned Conjugate Gradients.

1 Introduction

Recent work in Structure from Motion (SfM) has demonstrated the possibility of reconstructing geometry from large-scale community photo collections [1,2,3]. Bundle adjustment, the joint non-linear refinement of camera and point parameters, is a key component of most SfM systems, and one which can consume a significant amount of time for large problems. As the number of photos in such collections continues to grow into the hundreds of thousands or even millions, the scalability of bundle adjustment algorithms has become a critical issue.

The basic mathematics of the bundle adjustment problem are well understood [4], and there is also a freely available high-quality implementation – SBA [5]. SBA is based on a dense Cholesky factorization of the reduced camera matrix. It has space complexity that is quadratic and time complexity that is cubic in the number of photos. While this works well for problems with a few hundred photos, for problems involving tens of thousands of photos, it is prohibitively expensive.

⋆ Part of this work was done while the author was at University of Washington.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 29–42, 2010.

(a) Structured - 6375 photos (b) Unstructured - 4585 photos

Fig. 1. Connectivity graphs for a structured dataset (captured from a moving truck) and a community photo collection (consisting of photos matching the search term "Dubrovnik" downloaded from Flickr). For each dataset, we show an adjacency matrix representation of the connectivity graph, where black indicates a connection between two photos.

With the exception of a few efforts [6,7,8,1,9], the development of large-scale bundle adjustment algorithms has not received significant attention in the computer vision community. We believe this is because until now, the most common sources of large SfM problems have been video and structured survey datasets such as street-level and aerial imagery. For these datasets, the connectivity graph—i.e., the graph in which each photo is a node, and two photos are connected if they are looking at the same part of the scene—is extremely sparse, and has a mostly band-diagonal structure with a large diameter. For instance, in the case of data acquired using a camera mounted on a vehicle driving down a street, there is little to no overlap between photos taken even a few seconds apart. Figure 1(a) shows one such graph. Thus, techniques that reduce the size of the bundle adjustment problem by focusing on the most recently modified part of the reconstruction are quite effective [7,6].

Connectivity graphs of community photo collections are much less structured and have a significantly smaller diameter, as they tend to represent popular landmarks rather than a long, extended sequence of views. Figure 1(b) shows the graph for a set of photos of the city of Dubrovnik downloaded from Flickr. Compared to the structured dataset in Figure 1(a) which is 98% sparse with a mostly band diagonal structure, the graph for Dubrovnik is only 84% sparse, with a significantly more complex structure. This means that even though the dataset in Figure 1(a) has almost 1800 more photos than the dataset in Figure 1(b), the former requires 40x less time to find a sparse factorization of its reduced camera matrix than the latter.

In this paper, we present the design and implementation of a new inexact Newton type bundle adjustment algorithm, which uses substantially less time and memory than standard Schur complement based methods, without compromising on the quality of the solution. We explore the use of the Conjugate Gradients algorithm for calculating the Newton step and its performance as a function of some simple and computationally efficient preconditioners. We also show that the use of the Schur complement is not limited to factorization-based methods, how it can be used as part of the Conjugate Gradients (CG)

method without incurring the computational cost of actually calculating and storing it in memory, and how this use is equivalent to the choice of a particular preconditioner.

We present extensive experimental results on structured and unstructured datasets with a wide variety of problem complexity, and present recommendations based on these experiments. The code, test problems and detailed performance results from this paper are available at `http://grail.cs.washington.edu/projects/bal`.

The rest of the paper is organized as follow. We begin in Section 2 with a brief overview of the general nonlinear least squares problem, the Levenberg Marquardt (LM) algorithm, and the Schur complement trick. In Section 3, we introduce the inexact step LM algorithm, with a discussion of various methods for preconditioning the Conjugate Gradients (CG) algorithm in Section 4. Section 5 reports the results of our experiments and we conclude in Section 6 with a discussion.

2 Bundle Adjustment

Given a set of measured image feature locations and correspondences, the goal of bundle adjustment is to find 3D point positions and camera parameters that minimize the reprojection error. This optimization problem is usually formulated as a non-linear least squares problem, where the error is the squared L_2 norm of the difference between the observed feature location and the projection of the corresponding 3D point on the image plane of the camera. However, we are not limited to using the L_2 norm; even when robust loss functions like Huber's norm are used, the problem can be cast as a re-weighted non-linear least squares problem [10]. Thus in what follows, we will use the term bundle adjustment to mean a particular class of non-linear least squares problems.

2.1 Levenberg Marquardt Algorithm

The Levenberg-Marquardt (LM) algorithm [11] is the most popular algorithm for solving non-linear least squares problems, and the algorithm of choice for bundle adjustment. In this section, we begin with a quick review of LM, and then describe the Schur complement trick that substantially reduces the computational complexity of LM applied to bundle adjustment. Several excellent references exist for the reader interested in more details of LM [11,12,13,14].

Let $x \in \mathbb{R}^n$ be an n-dimensional vector of variables, and $F(x) = [f_1(x), \ldots, f_m(x)]^\top$ be a m-dimensional function of x. We are interested in solving the following optimization problem,

$$\min_x \frac{1}{2}\|F(x)\|^2 \ . \tag{1}$$

The Jacobian $J(x)$ of $F(x)$ is an $m \times n$ matrix, where $J_{ij}(x) = \partial_j f_i(x)$ and the gradient vector $g(x) = \nabla \frac{1}{2}\|F(x)\|^2 = J(x)^\top F(x)$.

The general strategy when solving non-linear optimization problems is to solve a sequence of approximations to the original problem [11]. At each iteration, the approximation is solved to determine a correction Δx to the vector x. For non-linear least squares, an approximation can be constructed by using the linearization $F(x + \Delta x) \approx F(x) + J(x)\Delta x$, which leads to the following linear least squares problem:

$$\min_{\Delta x} \frac{1}{2}\|J(x)\Delta x + F(x)\|^2 \tag{2}$$

Unfortunately, naïvely solving a sequence of these problems and updating $x \leftarrow x + \Delta x$ leads to an algorithm that may not converge. To get a convergent algorithm, we need to control the size of the step Δx. One way to do this is to introduce a regularization term:

$$\min_{\Delta x} \frac{1}{2}\|J(x)\Delta x + F(x)\|^2 + \mu\|D(x)\Delta x\|^2 \ . \tag{3}$$

Here, $D(x)$ is a non-negative diagonal matrix, typically the square root of the diagonal of the matrix $J(x)^\top J(x)$ and μ is a non-negative parameter that controls the strength of regularization. It is straightforward to show that the step size $\|\Delta x\|$ is inversely related to μ. LM updates the value of μ at each step based on how well the Jacobian $J(x)$ approximates $F(x)$. The quality of this fit is measured by the ratio of the actual decrease in the objective function to the decrease in the value of the linearized model $L(\Delta x) = \frac{1}{2}\|J(x)\Delta x + F(x)\|^2$. This kind of reasoning is the basis of Trust-region methods, of which LM is an early example [11].

The dominant computational cost in each iteration of the LM algorithm is the solution of the linear least squares problem (3). For general, small to medium scale least squares problems, the recommended method for solving (3) is using the the QR factorization [13]. However, the bundle adjustment problem has a very special structure, and a more efficient scheme for solving (4) can be constructed.

2.2 The Schur Complement Trick

We begin by introducing the regularized Hessian matrix $H_\mu(x) = J(x)^\top J(x) + \mu D(x)^\top D(x)$. It is easy to show that for $\mu D(x) > 0$, H_μ is a symmetric positive definite matrix and the solution to (3) can be obtained by solving the *normal equations*:

$$H_\mu(x)\Delta x = -g(x) \ . \tag{4}$$

Now, suppose that the SfM problem consists of p cameras and q points and the variable vector x has the block structure $x = [y_1, \ldots, y_p, z_1, \ldots, z_q]$. Where, y and z correspond to camera and point parameters, respectively. Further, let the camera blocks be of size c and the point blocks be of size s (for most problems $c = 6$–9 and $s = 3$).

In most cases, a key characteristic of the bundle adjustment problem is that there is no term f_i that includes two or more camera or point blocks. In other

words, each term $f_i(x)$ in the objective function can be re-written as $f_i(x) = f_i(y_{(i)}, z_{(i)})$, where, $y_{(i)}$ and $z_{(i)}$ are the camera and point blocks that occur in the i^{th} term. This in turn implies that the matrix H_μ is of the form

$$H_\mu = \begin{bmatrix} B & E \\ E^\top & C \end{bmatrix} , \tag{5}$$

where, $B \in \mathbb{R}^{pc \times pc}$ is a block diagonal matrix with p blocks of size $c \times c$ and $C \in \mathbb{R}^{qs \times qs}$ is a block diagonal matrix with q blocks of size $s \times s$. $E \in \mathbb{R}^{pc \times qs}$ is a general block sparse matrix, with a block of size $c \times s$ for each observation. Let us now block partition $\Delta x = [\Delta y, \Delta z]$ and $-g = [v, w]$ to restate (4) as the block structured linear system

$$\begin{bmatrix} B & E \\ E^\top & C \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} v \\ w \end{bmatrix} , \tag{6}$$

and apply Gaussian elimination to it. As we noted above, C is a block diagonal matrix, with small diagonal blocks of size $s \times s$. Thus, calculating the inverse of C by inverting each of these blocks is an extremely cheap, $O(q)$ algorithm. This allows us to eliminate Δz by observing that $\Delta z = C^{-1}(w - E^\top \Delta y)$, giving us

$$\left[B - EC^{-1}E^\top\right] \Delta y = v - EC^{-1}w . \tag{7}$$

The matrix

$$S = B - EC^{-1}E^\top , \tag{8}$$

is the Schur complement of C in H_μ. It is also known as the *reduced camera matrix*, because the only variables participating in (7) are the ones corresponding to the cameras. $S \in \mathbb{R}^{pc \times pc}$ is a block structured symmetric positive definite matrix, with blocks of size $c \times c$. The block S_{ij} corresponding to the pair of images i and j is non-zero if and only if the two images observe at least one common point.

Now, (6) can be solved by first forming S, solving for Δy, and then back-substituting Δy to obtain the value of Δz. Thus, the solution of what was an $n \times n$, $n = pc + qs$ linear system is reduced to the inversion of the block diagonal matrix C, a few matrix-matrix and matrix-vector multiplies, and the solution of block sparse $pc \times pc$ linear system (7). For almost all problems, the number of cameras is much smaller than the number of points, $p \ll q$, thus solving (7) is significantly cheaper than solving (6). This is the *Schur complement trick* [15].

This still leaves open the question of solving (7). The method of choice for solving symmetric positive definite systems exactly is via the Cholesky factorization [16] and depending upon the structure of the matrix, there are, in general, two options. The first is direct factorization, where we store and factor S as a dense matrix [16]. This method has $O(p^2)$ space complexity and $O(p^3)$ time complexity and is only practical for problems with up to a few hundred cameras. But, S is typically a fairly sparse matrix, as most images only see a small fraction of the scene. This leads us to the second option: sparse direct methods. These methods store S as a sparse matrix, use row and column re-ordering

algorithms to maximize the sparsity of the Cholesky decomposition, and focus their compute effort on the non-zero part of the factorization [17]. Sparse direct methods, depending on the exact sparsity structure of the Schur complement, allow bundle adjustment algorithms to significantly scale up over those based on dense factorization.

This however is not enough for community photo collections, where the size and sparsity structure of S (*e.g.* Figure 1) is such that even constructing it is a significant expense, and factoring it leads to near dense Cholesky factors. Hence we would like to find alternatives that do not depend on the construction, storage, and factorization of S and yet give good performance on large problems.

3 A Truncated Newton Solver

The factorization methods described above are based on computing an exact solution of (3). But it is not clear if an exact solution of (3) is necessary at each step of the LM algorithm to solve (1). In fact, we have already seen evidence that this may not be the case, as (3) is itself a regularized version of (2). Indeed, it is possible to construct non-linear optimization algorithms in which the linearized problem is solved approximately. These algorithms are known as inexact Newton or truncated Newton methods [11].

An inexact Newton method requires two ingredients. First, a cheap method for approximately solving systems of linear equations. Typically an iterative linear solver like the Conjugate Gradients method is used for this purpose [11]. Second, a termination rule for the iterative solver. A typical termination rule is of the form

$$\|H_\mu(x)\Delta x + g(x)\| \leq \eta_k \|g(x)\|. \tag{9}$$

Here, k indicates the LM iteration number and $0 < \eta_k < 1$ is known as the forcing sequence. Wright & Holt [18] prove that a truncated LM algorithm that uses an inexact Newton step based on (9) converges for any sequence $\eta_k \leq \eta_0 < 1$ and the rate of convergence depends on the choice of the forcing sequence η_k.

4 Preconditioned Conjugate Gradients

The convergence rate of CG for solving (4) depends on the distribution of eigenvalues of H_μ [19]. A useful upper bound is $\sqrt{\kappa(H_\mu)}$, where, $\kappa(H_\mu)$f is the condition number of the matrix H_μ. For most bundle adjustment problems, $\kappa(H_\mu)$ is high and a direct application of CG to (4) results in extremely poor performance.

The solution to this problem is to replace (4) with a *preconditioned* system. Given a linear system, $Ax = b$ and a preconditioner M the preconditioned system is given by $M^{-1}Ax = M^{-1}b$. The resulting algorithm is known as Preconditioned Conjugate Gradients algorithm (PCG) and its worst case complexity now depends on the condition number of the *preconditioned* matrix $\kappa(M^{-1}A)$.

The key computational cost in each iteration of PCG is the evaluation of the matrix vector product $\beta = A\alpha$ and solution of the linear system $M\phi = \psi$

for arbitrary vectors α and ψ. Thus, for each iteration of PCG to be efficient, M should be cheaply invertible and for the number of iterations of PCG to be small, the condition number $\kappa(M^{-1}A)$ should be as small as possible. The ideal preconditioner would be one for which $\kappa(M^{-1}A) = 1$. $M = A$ achieves this, but it is not a practical choice, as applying this preconditioner would require solving a linear system equivalent to the unpreconditioned problem. So how does one choose an effective preconditioner that is cheap to invert and results in a significant reduction of the condition number of the preconditioned matrix?

The simplest of all preconditioners is the diagonal or Jacobi preconditioner, *i.e.* , $M = \text{diag}(A)$, which for block structured matrices like H_μ can be generalized to the block Jacobi preconditioner. H_μ also has the special property that its diagonal blocks B and C are themselves block diagonal matrices. This property makes the block Jacobi preconditioner

$$M_J = \begin{bmatrix} B & 0 \\ 0 & C \end{bmatrix}. \tag{10}$$

the optimal block diagonal preconditioner for H_μ [20].

Another option is to apply PCG to the reduced camera matrix S instead of H_μ. One reason to do this is that S is a much smaller matrix than H_μ, but more importantly, it can be shown that $\kappa(S) \leq \kappa(H_\mu)$. There are two obvious choices for block diagonal preconditioners for S. The matrix B [21] and the block diagonal $\mathcal{D}(S)$ of S, *i.e.* the block Jacobi preconditioner for S.

Consider now, the generalized Symmetric Successive Over-relaxation (SSOR) preconditioner for H_μ,

$$M_\omega(P) = \begin{bmatrix} P & \omega E \\ 0 & C \end{bmatrix} \begin{bmatrix} P^{-1} & 0 \\ 0 & C^{-1} \end{bmatrix} \begin{bmatrix} P \\ \omega E^\top & C \end{bmatrix}, \tag{11}$$

where P is some easily invertible matrix and $0 \leq \omega < 2$ is a scalar parameter.

Observe that for $\omega = 0$, $M_0(B) = M_J$ is the block Jacobi preconditioner. More interestingly, for $\omega = 1$, it can be shown that using $M_1(P)$ as a preconditioner for H_μ is exactly equivalent to using the matrix P as a preconditioner for the reduced camera matrix S [19]. This means that for $P = I$ using $M_1(I)$ as a preconditioner for H_μ is the same as running pure CG on S and we can run PCG on S with preconditioners B and $\mathcal{D}(S)$ by using $M_1(B)$ and $M_1(\mathcal{D}(S))$ as preconditioners for H_μ. Thus, the Schur complement which started out its life as a way of specifying the order in which the variables should be eliminated from H_μ when solving (4) exactly, returns to the scene as a generalized SSOR preconditoner when solving the same linear system iteratively.

As discussed earlier, the cost of forming and storing the Schur complement S can be prohibitive for large problems. Indeed, for an inexact Newton solver that uses PCG on S, almost all of its time is spent in constructing S; the time spent inside the PCG algorithm is negligible in comparison.

Because PCG only needs access to S via its product with a vector, one way to evaluate Sx is to use (11) for $\omega = 1$. However we can do even better. Observe that,

$$x_1 = E^\top x,\ x_2 = C^{-1}x_1,\ x_3 = Ex_2,\ x_4 = Bx,\ Sx = x_4 - x_3\ . \tag{12}$$

Thus, we can run PCG on S with the same computational effort per iteration as PCG on H_μ, while reaping the benefits of a more powerful preconditioner. Even if we decide to use the block Jacobi preconditioner $\mathcal{D}(S)$, it can be constructed at a cost that is linear in the number of observations $O(m)$ and memory cost that is linear in the number of cameras - $O(p)$. Both of these are substantially less than the cost of computing and storing the full matrix S.

Equation (12) is closely related to *Domain Decomposition methods* for solving large linear systems that arise in structural engineering and partial differential equations. In the language of Domain Decomposition, each point in the SFM problem is a domain, and the cameras form the interface between these domains. The iterative solution of the Schur complement then falls within the sub-category of techniques known as Iterative Sub-structuring [19,22].

5 Experimental Evaluation

5.1 Algorithms

We compared the performance of six bundle adjustment algorithms: explicit-direct, explicit-sparse, normal-jacobi, explicit-jacobi, implicit-jacobi and implicit-ssor. The first two methods are exact step LM algorithms, and the remaining four are inexact step LM algorithms. explicit-direct, explicit-sparse and explicit-jacobi *explictly* construct the Schur complement matrix S and solve (7) using dense factorization, sparse direct factorization, and PCG using the block Jacobi preconditioner $\mathcal{D}(S)$ respectively. normal-jacobi uses PCG on H_μ with the block Jacobi preconditioner M_J. implicit-jacobi and implicit-ssor run PCG on S using the block Jacobi preconditioner $\mathcal{D}(S)$ and B respectively. Unlike explicit-jacobi they use (12) to *implicitly* evaluate matrix vector products with S.

Assuming that all the algorithms store H_μ in the same format, the difference between their memory usages depends on how they use the Schur complement S. implicit-jacobi, implicit-ssor and normal-jacobi do not compute or store S, and therefore require the least amount of memory. explicit-direct is the most expensive of the three as it uses $O(p^2)$ memory to store and factor S. explicit-sparse and explicit-jacobi are less expensive as they stores S as a sparse matrix, and thus their storage requirements scale with the sparsity of S. explicit-sparse requires additional storage to store the Cholesky factorization of S, and the amount of memory required is a function of the sparsity structure of S and not just the number of non-zero entries.

For each solver, LM was run for a maximum of 50 iterations, i.e. (3) was solved 50 times. After each LM iteration the step Δx may or may not be accepted, depending on whether it leads to a better solution. Inside each iteration of LM, PCG was run for a minimum of 10 iterations, and terminated when either $\|H_\mu(x)\Delta x + g(x)\| \leq \eta_k \|g(x)\|$ was satisfied or a 1000 iterations were performed. The forcing sequence η_k was set to a constant $\eta_k = 0.1$. At the beginning of LM, the square root of the diagonal of the matrix $J(x_0)^\top J(x_0)$ is estimated and used

as a scaling matrix for the variables. This is a standard method for normalizing all the variables in a problem [23] and is necessary as some parameters, (e.g., radial distortion), are up to 20 orders of magnitude more sensitive than others (e.g., rotation). For the factorization methods, especially CHOLMOD, this improves numerical stability. For the iterative solvers, this is equivalent to applying the Jacobi preconditioner before any of the other preconditioners are used.

All six algorithms were implemented as part of a single C++ code base. We use GotoBLAS2 [24] for dense linear algebra and CHOLMOD [17] for sparse Cholesky factorization. All experiments were performed on a workstation with dual Quad-core CPUs clocked at 2.27Ghz with 48GB RAM running a 64-bit Linux operating system.

5.2 Datasets

We experimented with two sources of data:

1. Images captured at a regular rate using a Ladybug camera mounted on a moving vehicle. Image matching was done by exploiting the temporal order of the images and the GPS information captured at the time of image capture.
2. Images downloaded from Flickr.com and matched by the authors of [3]. We used images from Trafalgar Square and the cities of Dubrovnik, Venice, and Rome.

For Flickr photographs, the matched images were decomposed into a skeletal set (i.e., a sparse core of images) and a set of leaf images [1]. The skeletal set was reconstructed first, then the leaf images were added to it via resectioning followed by triangulation of the remaing 3D points. The skeletal sets and the Ladybug datasets were reconstructed incrementally using a modified version of Bundler [25], which was instrumented to dump intermediate unoptimized reconstructions to disk. This gave rise to the Skeletal and the Ladybug problems. We refer to the bundle adjustment problems obtained after adding the leaf images to the skeletal set and triangulating the remaing points as the Final problems. For each dataset we use a nine parameter camera model (6 for pose, 1 for focal length and 2 for radial distortion).

Figure 2 plots the three types of problems. The x-axis is the number of images on a log-scale and the y-axis is the sparsity of the S matrix. The Ladybug (blue) set has small dense problems and large sparse problems with almost band diagonal sparsity. The Skeletal (red) set has small dense, and medium to large sparse problems with random sparsity. The Final (green) set has large problems with low to high sparsity. Their size and sparsity can pose significant challenges for state of the art algorithms. Complete details on the properties of each problem used in the experiments can be found on the project website.

5.3 Analysis

Detailed statistics on the performance of all algorithms are available on the project website. Here we summarize the broad trends in the data.

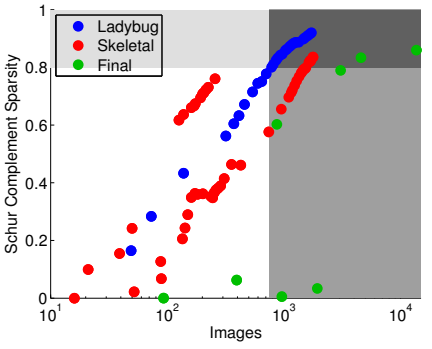


Fig. 2. *Datasets.* This scatter plot shows each of the datasets in our testbed, colored according to type (Ladybug, Skeletal, Final). The x-axis is the number of images in the problem and the y-axis is the sparsity of the Schur complement matrix S. The background of the plot is shaded according to the characteristics of the problem: small and dense (white), then in increasing gray-level, small and sparse, large and dense, and large and sparse.

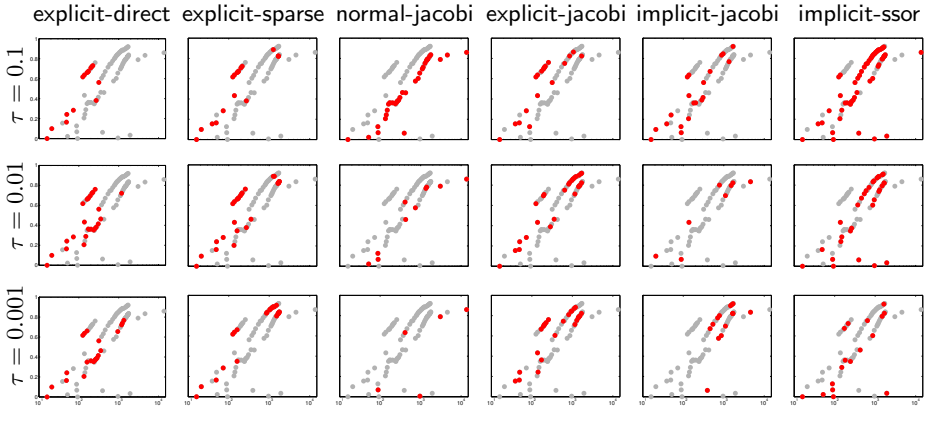


Fig. 3. *Performance analysis.* Each column in this set of plots corresponds to one of six algorithms, and each row corresponds to one of three tolerances τ. For each solver (column), a point is colored red if the solver was declared a winner for the given tolerance, and gray otherwise. Winnings solvers are the ones for which the relative decrease in the RMS error $(r_k - r^*)/(r_0 - r^*) \leq \tau$ in the least amount of time (there can be more than one such solver). The axes of the individual plots are the same as in Figure 2.

We compare solvers across problems by looking at how often they are the first one to improve the RMS error by a certain fraction. Concretely, for each solver and problem, let $r_k = \sqrt{\sum_i^m f_i^2(x_k)/m}$ denote the RMS error at end of iteration k and let r^* denote the minimum RMS error across all solvers for that problem. Then, for a given tolerance τ, we find the solvers for which $(r_k - r^*)/(r_0 - r^*) \leq \tau$ is satisfied in the least amount of time. We do this for three exponentially

tighter tolerances, $\tau = 0.1, 0.01, 0.001$. Figure 3 plots the results. The three rows correspond to the values of τ and the six columns correspond to different bundle adjustment algorithms. As in Figure 2, for each plot, x-axis is the number of images on a log scale, and y-axis is the sparsity of the Schur compliment matrix S. In each plot, we plot all the problems in light grey, and then in red, the problems for which that solver at that tolerance level was one of the winners[1].

From Figure 3, we observe that for problems with up to a few hundred images and all three tolerances, explicit-direct offers consistently good performance. State of the art BLAS and LAPACK libraries on multicore systems have excellent performance, and for small to moderate sized matrices, an exact step LM with a dense Cholesky solver is hard to beat. This explains the continuing popularity and success of SBA [5].

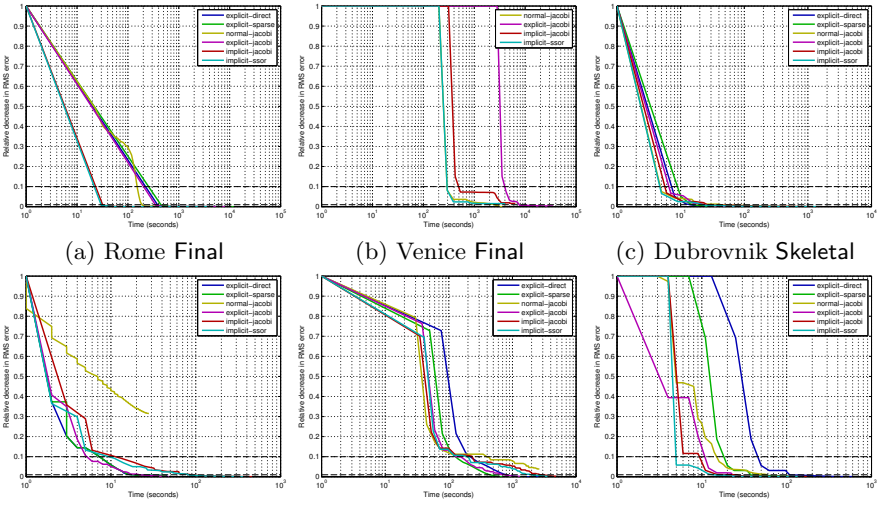
For larger problems and high tolerance values $\tau = 0.1$, both normal-jacobi and implicit-ssor do well, with implicit-ssor working on a much wider variety of problems. As the value τ decreases, the performance of normal-jacobi rapidly degrades, indicating that the quality of preconditioning is not good enough to produce high quality Newton steps in a short amount of time. On the other hand as τ is decreased, explicit-jacobi which is the most expensive of the iterative solvers, becomes a viable candidate with the block Jacobi preconditioning of S starting to show its benefits. implicit-ssor beats explicit-jacobi when S has low sparsity. This is not surprising, as the cost of computing a nearly dense reduced camera matrix becomes a significant factor, where as implicit-ssor is able to avoid this extra computational burden.

A closer examination of the data reveals that despite an overall degradation in performance, normal-jacobi continues to work well for the larger problems in the Final set. We believe this is because of the structure of the Skeletal sets algorithm. After the skeletal set has been reconstructed, the geometric core of the reconstruction is quite rigid and stable. The error in the reconstruction after the leaf images have been added is mostly local and no major global changes that span the entire reconstruction are expected. Therefore, the simple block Jacobi preconditioner captures the structure of H_μ quite well and at a substantially less computational cost than any other preconditioner.

It is also worth observing that for some of the problems, as the value of τ is decreased, factorization-based solvers become more competitive. This is expected, as lower values of τ demand that the LM algorithm take higher quality steps at each iteration. In this regime, the higher cost of the exact step algorithms, at least for the smaller problems, wins over the increased iteration complexity of the inexact step algorithms. Better performance for inexact step algorithms will require more sophisticated forcing sequence η_k and preconditioners.

There were two surprises. First, the discrepancy in the performance of explicit-jacobi and implicit-jacobi . In exact arithmetic, these two algorithms should return exactly the same answer, but that is not the case in practice. A closer look at the data revealed that for the same linear system, the two methods resulted in different number of iterations and answers, sometimes significantly so, indicating

[1] Since time is measured in seconds, there may be more than one such solver.



(a) Rome Final (b) Venice Final (c) Dubrovnik Skeletal

(d) Trafalgar Skeletal (e) Venice Skeletal Ladybug

Fig. 4. A sampling of run time plots. In each plot, the x-axis is time on a log scale, and the y-axis is the relative decrease in the RMS error $(r_k - r^*)/(r_0 - r^*)$. The three black dashed horizontal lines in each plot correspond to the three tolerances, *i.e.* , $\tau = 0.1, 0.01$ and 0.001. Note that explicit-direct and explicit-sparse are missing from the Venice Final plot as they ran out of memory.

numerical instability in implicit-jacobi which merits further investigation. Second, explicit-sparse did not emerge as a clear winner in any of the problem categories. Either the problems were too small for the additional setup cost and the more complicated algorithm used in CHOLMOD to beat dense Cholesky factorization, or the problems were large enough that the exact factorization algorithms, sparse and dense, were beaten by the inexact step algorithms.

In summary, we observe that for large scale problems, the iterative methods are a significant memory and time win over Cholesky factorization-based methods. Particularly for Final problems, this can be the difference between being able to solve the problem or not, as evidenced by the large Venice example. But even for medium sized problems involving a few thousand images, the iterative solvers are up to an order of magnitude faster while consuming 3-5 times less memory. For the sparse problems in the Ladybug and Skeletal datasets, the advantage is usually in terms of memory and simplicity of implementation rather than time, as the cost of exact factorization is offset by its superior quality. However, we must remember that these experiments were performed on state of the art workstations with much more RAM than is commonly available today, which makes the memory usage of the iterative methods even more attractive.

For small to medium problems, we recommend the use of a dense Cholesky-based LM algorithm. For larger problems, the situation is more complicated and there is no one clear answer. Both implicit-ssor and explicit-jacobi offer competitive solvers, with implicit-ssor being preferred for problems with lower sparsity and

explicit-jacobi for problems with high sparsity. We hope that once the cause of numerical instability in implicit-jacobi can be understood and rectified, it will offer a memory efficient solver that bridges the gap between these two solvers and works on large bundle adjustment problems, independent of their sparsity.

6 Discussion

The classical solution to bundle adjustment is based on exploiting the primary sparsity structure of the problem to form a Schur complement and factoring it [26,4,10]. With the exception of a few recent attempts [27,28], it has remained the dominant method for doing bundle adjustment. While suitable for problems with a few hundred images, this method does not scale to larger problems with thousands of images. In this paper, we have shown with the help of an extensive test suite of large scale bundle adjustment problems that a truncated Newton style LM algorithm coupled with a simple preconditioner delivers state of the art performance at a fraction of the time and memory cost of methods based on factoring the Schur complement.

Going forward, the preconditioners considered in this paper are relatively simple but we hope that the identification with domain decomposition methods opens up the possibility of using much more sophisticated preconditioners developed in the structural engineering literature [19,22]. Numerical stability is another critical issue. As we noted earlier, even though explicit-jacobi and implicit-jacobi are algebraically equivalent algorithms, they show problem-dependent numerical behavior. A more thorough development that accounts for the numerical stability of evaluating the matrix-vector products using the explicit and the implicit schemes is needed.

Acknowledgements. The authors are grateful to Drew Steedly and David Nister for useful discussions and Kristin Branson for reading multiple drafts of the paper.

This work was supported in part by National Science Foundation grant IIS-0811878, SPAWAR, the Office of Naval Research, the University of Washington Animation Research Labs, Google, Intel, and Microsoft.

References

1. Snavely, N., Seitz, S.M., Szeliski, R.: Skeletal graphs for efficient structure from motion. In: CVPR, pp. 1–8 (2008)
2. Li, X., Wu, C., Zach, C., Lazebnik, S., Frahm, J.: 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)
3. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building Rome in a day. In: ICCV (2009)
4. Triggs, B., McLauchlan, P., Hartley, R.I., Fitzgibbon, A.: Bundle Adjustment - A modern synthesis. In: Vision Algorithms 1999, pp. 298–372 (1999)

5. Lourakis, M., Argyros, A.A.: SBA: A software package for generic sparse bundle adjustment. TOMS 36, 2 (2009)
6. Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., Sayd, P.: Generic and real-time structure from motion using local bundle adjustment. Image and Vision Computing 27, 1178–1193 (2009)
7. Steedly, D., Essa, I.: Propagation of innovative information in non-linear least-squares structure from motion. In: ICCV, pp. 223–229 (2001)
8. Ni, K., Steedly, D., Dellaert, F.: Out-of-core bundle adjustment for large-scale 3d reconstruction. In: ICCV (2007)
9. Steedly, D., Essa, I., Dellaert, F.: Spectral partitioning for structure from motion. In: ICCV, pp. 996–1003 (2003)
10. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2003)
11. Nocedal, J., Wright, S.: Numerical optimization. Springer, Heidelberg (2000)
12. More, J.: The Levenberg-Marquardt algorithm: implementation and theory. Lecture Notes in Math. 630, 105–116 (1977)
13. Björck, A.: Numerical methods for least squares problems. SIAM, Philadelphia (1996)
14. Madsen, K., Nielsen, H., Tingleff, O.: Methods for non-linear least squares problems (2004)
15. Brown, D.C.: A solution to the general problem of multiple station analytical stereotriangulation. Technical Report 43, Patrick Airforce Base, Florida (1958)
16. Trefethen, L., Bau, D.: Numerical linear algebra. SIAM, Philadelphia (1997)
17. Chen, Y., Davis, T., Hager, W., Rajamanickam, S.: Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate. TOMS 35 (2008)
18. Wright, S.J., Holt, J.N.: An inexact Levenberg-Marquardt method for large sparse nonlinear least squares. J. Austral. Math. Soc. Ser. B 26, 387–403 (1985)
19. Saad, Y.: Iterative methods for sparse linear systems. SIAM, Philadelphia (2003)
20. Elsner, L.: A note on optimal block-scaling of matrices. Numer. Math. 44, 127–128 (1984)
21. Mandel, J.: On block diagonal and Schur complement preconditioning. Numer. Math. 58, 79–93 (1990)
22. Mathew, T.: Domain decomposition methods for the numerical solution of partial differential equations. Springer, Heidelberg (2008)
23. Dennis Jr., J., Gay, D., Welsch, R.: Algorithm 573: NL2SOLan adaptive nonlinear least-squares algorithm [E4]. TOMS 7, 369–383 (1981)
24. Goto, K., Van De Geijn, R.: High-performance implementation of the level-3 blas. TOMS 35, 1–14 (2008)
25. Snavely, N., Seitz, S.M., Szeliski, R.: Photo Tourism: Exploring photo collections in 3D. TOG 25, 835–846 (2006)
26. Engels, C., Stewenius, H., Nister, D.: Bundle adjustment rules. Photogrammetric Computer Vision 2 (2006)
27. Lourakis, M., Argyros, A.: Is Levenberg-Marquardt the most efficient optimization algorithm for implementing bundle adjustment. In: ICCV, pp. 1526–1531 (2005)
28. Byröd, M., Åström, K., Lund, S.: Bundle adjustment using conjugate gradients with multiscale preconditioning (2009)

Sparse Non-linear Least Squares Optimization for Geometric Vision

Manolis I.A. Lourakis

Institute of Computer Science, Foundation for Research and Technology - Hellas
N. Plastira 100, Vassilika Vouton, Heraklion, Crete, 700 13 Greece
http://www.ics.forth.gr/~lourakis/sparseLM/

Abstract. Several estimation problems in vision involve the minimization of cumulative geometric error using non-linear least-squares fitting. Typically, this error is characterized by the lack of interdependence among certain subgroups of the parameters to be estimated, which leads to minimization problems possessing a sparse structure. Taking advantage of this sparseness during minimization is known to achieve enormous computational savings. Nevertheless, since the underlying sparsity pattern is problem-dependent, its exploitation for a particular estimation problem requires non-trivial implementation effort, which often discourages its pursuance in practice. Based on recent developments in sparse linear solvers, this paper provides an overview of `sparseLM`, a general-purpose software package for sparse non-linear least squares that can exhibit arbitrary sparseness and presents results from its application to important sparse estimation problems in geometric vision.

1 Introduction

A plethora of estimation problems in multiple view geometry employ model fitting to infer mathematical objects from image data. Fitting is accomplished by minimizing the total geometric error pertaining to overdetermined sets of image measurements, which is an approach that has proven to constitute a major contributor to the success of contemporary algorithms in multiple view geometry [1]. The total geometric error is expressed by a sum-of-squares cost function (i.e., a L_2 norm), whose minimizer represents the statistically optimal estimate of the sought objects under Gaussian noise. Owing to their non-convexity, L_2 cost functions are minimized with iterative non-linear least squares techniques, of which the Levenberg-Marquardt (LM) algorithm has become the de facto standard. LM operates by repeatedly linearizing the function to be minimized in the neighborhood of the current minimizer estimate and computing an improvement to it through the solution of a linear system defined with the aid of the Jacobian and known as the *normal equations*. Considering that each computation of the solution to a dense linear system has complexity $O(N^3)$ in the number of unknown parameters, it is clear that general purpose LM implementations are computationally very demanding when employed to minimize functions involving a large number of parameters N.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 43–56, 2010.

Fortunately, when dealing with large estimation problems arising in multiple view geometry, the corresponding geometric error exhibits lack of interdependence among certain subgroups of the parameters to be estimated. This observation translates to Jacobians for the least squares minimization that are sparse, that is, consist of mostly zero elements. In turn, sparse Jacobians yield normal equation systems with sparse block structure. Examples of such sparse problems include single view reconstruction [2], homography, fundamental matrix and trifocal tensor estimation with the "Gold Standard" algorithms [1] (pp.114, 285 & 397 resp.), mosaicking [3] and bundle adjustment [4,5]. It is well-known that by avoiding storing and operating on zero elements of the normal matrix during the course of LM, substantial memory and execution time benefits can be gained. For instance, Appendix 6 of [1] describes a scheme for effectively dealing with the commonly encountered "arrowhead" type of sparseness (see also Fig. (1)(a)). This scheme performs a partitioning of the set of parameters in two functionally distinct groups and solves the normal equations by employing the corresponding Schur complement of the normal matrix. Its adoption has facilitated the implementation of LM variants tailored to the problem of bundle adjustment that divide the normal matrix into camera and structure blocks and are capable of successfully dealing with large reconstruction problems [5]. Despite its usefulness, the aforementioned scheme is not suited to all sparse problems that might be encountered in multiple view geometry, while its implementation is problem-specific and rather complicated. Therefore, considerable effort is required for developing LM variants customized to a particular sparse problem, making the latter task to be perceived as a daunting endeavor by both vision researchers and practitioners.

The reason behind the lack of universal applicability of the partitioning scheme of [1] is that its assumption of only two functional groups of parameters is not valid for all estimation problems. In other words, there exist problems whose Jacobian (and, therefore, normal equations) sparsity pattern has a more complex structure (e.g. Fig. 1(b)). Nonetheless, if an effective mechanism of dealing with arbitrary sparseness is available, then all sparse geometric vision estimation problems can be cast as special cases of the general sparse non-linear least squares minimization problem. During the last few years, such mechanisms have emerged in the form of a number of algorithms and corresponding implementations for the direct solution of large sparse linear systems of equations [6]. Compared to iterative methods [7], sparse direct methods do not employ preconditioners, do not suffer from slow convergence, produce exact rather than approximate solutions and their technology is well developed. Thus, they are more general and robust, therefore better suited as general-purpose linear solvers.

This work builds upon existing direct sparse solvers and employs them for developing `sparseLM`, a package fulfilling the need for a quality software designed for general-purpose, arbitrarily sparse non-linear least squares fitting. `sparseLM` is implemented in C and its source code is publicly available under the GNU GPL. To the best of the author's knowledge, no other comparable software is currently freely available with an open source license. Brief introductions to the LM algorithm and sparse direct solvers are supplied in sections 2 and 3, respectively. Section 4 presents

the major design guidelines and implementation issues related to `sparseLM`. Experimental results from the application of `sparseLM` to practical vision problems are provided in section 5 and the paper concludes in section 6.

2 The Levenberg-Marquardt Algorithm

The LM algorithm is an iterative technique that locates a local minimum of a multivariate function that is expressed as the sum of squares of non-linear real-valued functions. For the sake of completeness, a short description of the LM algorithm is provided next. However, a detailed analysis of the LM algorithm is beyond the scope of this paper and the interested reader is referred elsewhere [8].

Let f be an assumed functional relation which maps a *parameter vector* $\mathbf{p} \in \mathcal{R}^m$ to an estimated *measurement vector* $\hat{\mathbf{x}} = f(\mathbf{p}),\ \hat{\mathbf{x}} \in \mathcal{R}^n$. An initial parameter estimate $\mathbf{p}_0$ and a measured vector $\mathbf{x}$ are provided and it is desired to find the vector $\mathbf{p}^+$ that best satisfies the functional relation f locally, i.e. minimizes the squared distance $\epsilon^T\epsilon$ with $\epsilon = \mathbf{x} - \hat{\mathbf{x}}$ for all $\mathbf{p}$ within a m-sphere having a small radius. The basis of the LM algorithm is a linear approximation to f in the neighborhood of $\mathbf{p}$. Denoting by $\mathbf{J}$ the Jacobian matrix $\frac{\partial f(\mathbf{p})}{\partial \mathbf{p}}$, a Taylor series expansion for a small $||\delta_\mathbf{p}||$ leads to the following approximation:

$$f(\mathbf{p} + \delta_\mathbf{p}) \approx f(\mathbf{p}) + \mathbf{J}\delta_\mathbf{p}. \tag{1}$$

Like all non-linear optimization methods, LM is iterative: Initiated at the starting point $\mathbf{p}_0$, it produces a series of vectors that converge towards a local minimizer $\mathbf{p}^+$ for f. Hence, at each iteration, it is required to find the step $\delta_\mathbf{p}$ that minimizes the quantity

$$||\mathbf{x} - f(\mathbf{p} + \delta_\mathbf{p})|| \approx ||\mathbf{x} - f(\mathbf{p}) \ - \ \mathbf{J}\delta_\mathbf{p}|| = ||\epsilon - \mathbf{J}\delta_\mathbf{p}||. \tag{2}$$

Thus, the sought $\delta_\mathbf{p}$ is obtained from a linear least-squares problem which is solved using the normal equations:

$$\mathbf{J}^T\mathbf{J}\delta_\mathbf{p} = \mathbf{J}^T\epsilon. \tag{3}$$

An alternative to minimizing (2) employs the QR decomposition, which is nevertheless up to a factor of two slower than the normal equations (cf. [4], p.315). Matrix $\mathbf{J}^T\mathbf{J}$ in Eq. (3) is the first order approximation to the Hessian of $\frac{1}{2}\epsilon^T\epsilon$ [8], whereas $\delta_\mathbf{p}$ is the *Gauss-Newton* step. The LM algorithm actually solves a slight variation of Eq. (3), known as the *augmented normal equations*

$$\mathbf{N}\delta_\mathbf{p} = \mathbf{J}^T\epsilon, \text{ with } \mathbf{N} \equiv \mathbf{J}^T\mathbf{J} + \mu\mathbf{I} \text{ and } \mu > 0, \tag{4}$$

where $\mathbf{I}$ is the identity matrix. The strategy of altering the diagonal elements of $\mathbf{J}^T\mathbf{J}$ is called *damping* and μ is a regularization parameter referred to as the *damping term*. If the updated parameter vector $\mathbf{p} + \delta_\mathbf{p}$ with $\delta_\mathbf{p}$ computed from Eq. (4) leads to a reduction in the error $\epsilon^T\epsilon$, the update is accepted and the process repeats with a decreased damping term. Otherwise, the damping term is increased,

the augmented normal equations are solved again and the process iterates until a value of $\delta_{\mathbf{p}}$ that decreases the error is found. The process of repeatedly solving Eq. (4) for different values of the damping term until an acceptable update to the parameter vector is found corresponds to one iteration of the LM algorithm.

The damping term is adaptively adjusted at each iteration of LM to assure a reduction in $\epsilon^T\epsilon$. By doing so, LM is capable of alternating between a slow descent approach when being far from the minimum and a fast convergence when being in the minimum's neighborhood: If the damping is set to a large value, matrix $\mathbf{N}$ in Eq. (4) is nearly diagonal and the LM update step $\delta_{\mathbf{p}}$ is near the steepest descent direction $\mathbf{J}^T\epsilon$. Moreover, the magnitude of $\delta_{\mathbf{p}}$ is reduced, ensuring that excessively large Gauss-Newton steps are not taken. A large damping term also handles situations where the Jacobian is rank deficient and $\mathbf{J}^T\mathbf{J}$ is therefore singular. The damping term can be chosen so that the symmetric matrix $\mathbf{N}$ in Eq. (4) is non-singular and, therefore, positive definite (SPD), ensuring that the $\delta_{\mathbf{p}}$ computed from it is a descent direction. In this way, LM can defensively navigate a region of the parameter space in which the model is highly non-linear. If, on the other hand, the damping is small, the LM step approximates the exact Gauss-Newton step, lending LM rapid convergence.

3 Direct Sparse Linear Solvers

The solution of systems of sparse linear equations lies at the crux of numerous computational problems. Direct methods for solving the linear system $\mathbf{Ax} = \mathbf{b}$, where the coefficient matrix $\mathbf{A}$ is sparse, involve the explicit factorization of a suitable permutation of $\mathbf{A}$ into the product of lower and upper triangular matrices $\mathbf{L}$ and $\mathbf{U}$. If $\mathbf{A}$ is symmetric and, further, positive definite, $\mathbf{U} = \mathbf{L}^T$ (i.e., Cholesky factorization); in the indefinite case $\mathbf{U} = \mathbf{DL}^T$, where $\mathbf{D}$ is block diagonal. Forward elimination followed by backward substitution completes the solution procedure for the right-hand side $\mathbf{b}$. The main complication when developing direct solvers for sparse matrices stems from the requirement to efficiently handle *fill-in*, i.e. limit the number of elements which change from an initial zero in the permuted $\mathbf{A}$ to a non-zero value in the factors $\mathbf{L}$ and $\mathbf{U}$.

Several algorithms and corresponding software codes implementing direct methods have appeared in recent years. Despite their individual peculiarities, sparse direct solvers operate in distinct phases, outlined as follows [9,6]:

1. An ordering phase that permutes rows and columns to ensure either that the factors will suffer little fill-in or to yield a matrix with special structure (e.g. block triangular). The choice of an ordering algorithm is crucial to the efficiency of any direct solver. Since computing an optimal ordering is NP-complete, various heuristics are used in practice [10,11].
2. An analysis or symbolic factorization phase concerned with analyzing the matrix's structure to determine a pivot sequence (optional) and the non-zero structures of the factors. A good pivot sequence should significantly reduce the memory requirements as well as the floating point operations count. Occasionally, this phase is combined with the ordering one.

3. A numerical factorization phase that uses the pivot sequence to factorize the matrix.
4. A solve phase that performs forward elimination followed by back substitution using the computed factors.

The first two phases are independent of the matrix's numerical values and depend only on its non-zero structure. For SPD matrices, the pivot sequence may be chosen based solely on the sparsity pattern, therefore the analysis phase involves no computation on real numbers. When implemented serially, the factorization is typically the most time-consuming of the different phases whereas the solve phase is generally significantly faster. Performance can be accelerated with parallel processing, employing the MPI-based implementations for distributed memory architectures that are available for some of the solvers. Another potentially useful feature of some implementations is their ability to work out-of-core, i.e. to hold the coefficient matrix and/or its factor in disk files, thereby substantially reducing the amount of main memory required by the solver and enabling it to tackle larger problems.

4 Implementation Issues

This section discusses several choices made during the design of `sparseLM` with the twofold objective of maximizing its performance while shielding the user from the algorithmic details associated with direct solvers. Since the optimization aspects of `sparseLM` are more or less standard, the emphasis is on sparseness and means of better taking advantage of it.

4.1 Sparse Matrix Formats

We start with a short description of general storage formats for sparse matrices. These formats make no assumptions regarding the sparsity structure and store non-zero elements by allocating contiguous memory storage for them along with some additional index information for keeping track where they fit into the full matrix. The Compressed Row Storage (CRS) format stores non-zero elements in row-major order, whereas Compressed Column Storage (CCS) adopts column-major ordering. More details can be found in [12].

4.2 Jacobian Representation and Computation

From a user's perspective, the provision of derivatives is one of the most bewildering practical aspects of non-linear least squares solvers. In the case of `sparseLM`, the Jacobian has been further assigned the role of specifying the sparsity pattern of the problem at hand: Its element at position (i, j) is non-zero if and only if measurement i depends upon variable j. In other words, the Jacobian can be thought of as a parameter - observation connection graph prescribing which (parameter, observation) pairs have direct interaction. `sparseLM` accepts Jacobians in either CRS or CCS format, allowing user applications to choose

the representation that is most natural to them. Jacobians can be hand-coded by the user or, more conveniently, generated through the use of automatic differentiation tools that work by systematically applying the chain rule to a given code segment. Additionally, `sparseLM` offers the possibility of numerically approximating the Jacobian using forward finite differences on data provided by successive invocations of f (cf. Eq (1)). In that case, only the sparsity pattern of the Jacobian should be specified by the user, whereas its numerical values are approximated by `sparseLM`. To reduce the total number of invocations, the Jacobian is approximated using a scheme that computes several of its columns with a single evaluation of f, exploiting its sparse structure as explained in [8], ch. 7. For a m-dimensional parameter vector, this scheme requires much fewer evaluations than the $m + 1$ ones that would be required by the naive approach of computing a single column of the Jacobian per evaluation of f. However, considering that they lead to faster convergence, analytic Jacobians should be preferred over approximated ones whenever possible.

4.3 Approximate Hessian Computation

A key aspect of `sparseLM`'s implementation concerns the efficient computation of the first order approximation to the Hessian, i.e. of matrix $\mathbf{J}^T\mathbf{J}$ in Eq (4). $\mathbf{J}^T\mathbf{J}$ is stored internally in the CCS format since this is the one most frequently employed among the implementations of direct sparse solvers. Multiplication of sparse matrices is considerably more challenging than that of dense ones, since the sparsity pattern of the product should first be discovered and then the operations for calculating the product's non-zeros should be carried out in a manner efficient with respect to the matrices memory storage format. An important observation concerning the sparsity pattern of $\mathbf{J}^T\mathbf{J}$ is that it does not change among LM iterations. Therefore, `sparseLM` makes its computation more efficient by computing its non-zero structure only once ignoring numerical cancellation and then reusing it when evaluating an actual product. Another performance improvement stems from exploiting symmetry. Thus, `sparseLM` computes only the lower triangular part of $\mathbf{J}^T\mathbf{J}$ and then copies it to the upper half, effectively reducing the number of computations roughly in half. In fact, even the copying operation can be skipped for some of the solvers since those that are designed for symmetric systems access only the triangular part of the coefficient matrix. Depending on whether the Jacobian $\mathbf{J}$ is supplied in CRS or CCS format, the product $\mathbf{J}^T\mathbf{J}$ is formed by an efficient technique that traverses $\mathbf{J}$ in a row-wise or column-wise fashion, respectively, ensuring that the pattern of accesses to its elements matches their physical layout in memory.

4.4 Choice of Linear Solver

As in the case of dense linear systems, it is generally advantageous in terms of performance to employ a direct sparse solver whose prerequisites closely match the intrinsic properties of the problem at hand. In the context of sparse non-linear least squares, the augmented normal equations matrix of Eq. (4) is SPD, thus

the direct solvers of choice are those designed to perform sparse Cholesky factorization. Still, more general solvers targeted to indefinite or even non-symmetric systems are clearly also usable. Advanced features such as provision for parallel or out-of-core processing should also be taken into consideration. Comparative evaluations of direct solvers in the literature indicate that no single one is universally the best [9]. For this reason, `sparseLM` includes interfaces to a wide variety of codes, the list of which currently consists of `LDL` [13], HSL's `MA57/MA47/MA27` [14], `PARDISO` [15], `SuperLU` [16], `TAUCS` [17], `UMFPACK` [18], `CSparse` [6], `CHOLMOD` [19] `SPOOLES` [20], and `MUMPS` [21]. Moreover, `sparseLM` has been designed so that expanding this list with more solvers in the future is straightforward. `CHOLMOD` [19], a set of routines for factorizing sparse SPD matrices, is used as `sparseLM`'s default solver. Regarding ordering, `CHOLMOD` automatically chooses between approximate minimum degree (AMD) [10] and graph-based nested dissection (METIS) [11]. Its overall performance was found to be quite competitive by the recent survey of Gould et al. [9].

Independently of the choice of a direct solver, its application in the context of the LM algorithm can be made more efficient by the following observation: During the course of the LM algorithm, several linear systems with identical sparsity patterns are repeatedly solved. Thus, as explained in section 3, the corresponding symbolic factorization is computed only once and then reused for numerically solving all subsequent linear systems.

5 Experimental Results

This section provides an experimental evaluation of `sparseLM`, applying it to three important problems in multiple view geometry and comparing its performance against alternative established approaches. The problems in question are bundle adjustment, trifocal tensor and homography estimation.

5.1 Euclidean Bundle Adjustment

In this section two sets of experiments are conducted, aiming at comparing the performance of `sba` [5] against that of `sparseLM` applied to Euclidean sparse Bundle Adjustment (BA). `sba` is our freely available package for BA that implements the partitioning scheme of [1] to solve the sparse augmented equations. It is heavily optimized and provides increased flexibility by allowing user-defined parameterizations for cameras and points as well as projection functions, thus being able to support a wide range of manifestations of the multiple view reconstruction problem. Being custom-written to match the sparsity structure of the BA problem, `sba` is expected to generally excel in performance. Nevertheless, it is instructive to examine when this conjecture holds and how close are the performances of the two approaches.

The first set of experiments relies on the eight test sequences also employed in [5]. Each experiment involves a set of 3D points whose image projections have been identified in a number of real images acquired by an intrinsically calibrated

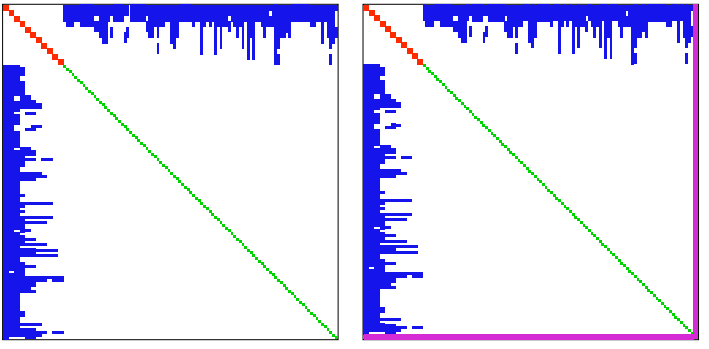
Table 1. Statistics for Euclidean BA using the `sparseLM` and `sba` packages: Total number of images, total number of variables, total number of objective function/Jacobian evaluations, total number of iterations and linear systems solved, elapsed execution time in seconds. Identical values for the user-defined minimization parameters have been used throughout all experiments.

Sequence	# imgs	# vars	func/jac evals		iter./sys. solved		exec. time	
			sparseLM	sba	sparseLM	sba	sparseLM	sba
"movi"	59	5688	20/18	20/20	18/20	20/20	4.26	3.69
"sagalassos"	26	5283	41/33	40/30	33/41	30/40	6.55	3.95
"arenberg"	22	4137	22/15	25/17	15/22	17/25	3.89	2.68
"basement"	11	981	33/22	32/23	22/33	23/32	0.57	0.28
"house"	10	1605	24/17	27/20	17/24	20/27	0.73	0.38
"maquette"	54	15945	29/21	30/23	21/29	23/30	13.20	7.98
"desk"	46	10542	28/20	32/22	20/28	22/32	8.51	6.06
"calgrid"	27	2328	25/19	21/20	19/25	20/21	15.61	8.58

moving camera. Estimates of the Euclidean 3D structure and camera motions have been computed using a sequential structure and motion estimation technique. Those estimates serve as starting points for bootstrapping refinements that are based on BA using `sba` and `sparseLM`. Camera motions corresponding to all but the first frame are defined relative to the initial camera location. The former is taken to coincide with the employed world coordinate frame. Camera rotations are parameterized by quaternions while translations and 3D points by 3D vectors.

Table 1 illustrates several statistics gathered from the application of `sba` and `sparseLM`-based Euclidean BA to the eight test sequences. Each row corresponds to a single sequence and columns are as follows: The first column corresponds to the total number of images that were employed in BA. The second column is the total number of motion and structure variables pertaining to the minimization. The third column shows the total number of objective function/Jacobian evaluations during BA for both approaches. The number of iterations needed for convergence and the total number of linear systems that were solved are shown in the fourth column. The last column shows the time in seconds elapsed during execution of BA. All experiments were conducted on an Intel P4@1.8 GHz running Linux and unoptimized BLAS. Both approaches converged to the same solutions for each sequence, therefore the corresponding final reprojection errors are not reported. As it is evident from the last column, BA with the aid of `sparseLM` is roughly at most two times slower than that employing `sba`. This is a remarkable result showing that the increased generality of `sparseLM` does not come at the price of performance.

At this point, it is enlightening to point out a few limitations of `sba` that are removed by the `sparseLM`-based approach to BA. `sba` assumes no coupling among the parameters for different cameras or different points. While this assumption is valid in many cases, there exist some situations where it imposes insurmountable restrictions. One such situation is illustrated in Fig. (1)(b) and concerns a sequence acquired with a camera having constant intrinsics that are to be refined



(a) (b)

Fig. 1. Visualization of the approximate Hessian's structure for two BA problems involving the "basement" sequence. (a) is 366×366 and arises in BA for camera motion and structure parameters (arranged in that order), (b) is 371×371 and corresponds to BA for camera motion, structure and constant across frames intrinsic parameters. Colored dots correspond to non-zero elements with red arising from motion-motion parameter pairs, green from point-point pairs, blue from motion-point pairs and magenta from motion-intrinsic, point-intrinsic and intrinsic-intrinsic pairs. `sba` cannot handle (b) due to the horizontal and vertical non-zero bands (in magenta) induced at its bottommost and rightmost parts by the sharing of intrinsic parameters. To improve the readability of graphs, only the first 100 points have been included in the BA.

via BA. In this case, the intrinsic calibration parameters must be shared by all images, violating `sba`'s assumption of independent camera parameters. Other examples involve the cases of employing inter-feature measurements such as distances or angles between points, coplanarity constraints on subgroups of points, articulated motion, etc. Another limitation stems from `sba`'s current implementation, which when forming the reduced bundle system assumes a dense structure for the Schur complement of the points submatrix in the approximate Hessian (i.e. the block matrix $\mathbf{S}$ in p.2:13 of [5]). This matrix, whose ij block is zero if images i and j have no points in common, is factored with a dense Cholesky decomposition to update the camera parameters. While it is reasonable to expect that for small problems such as the ones employed here most features are seen in all images and, therefore, matrix $\mathbf{S}$ is dense[1], for larger, more loosely connected image sets where each image only sees a small fraction of the features, $\mathbf{S}$ can become quite sparse. BA using `sparseLM` does not suffer from any of the aforementioned shortcomings since it treats the Jacobian (and therefore the Hessian) as a matrix with arbitrary sparseness, not needing to compute and factor $\mathbf{S}$.

To study the effect on performance of the density of matrix $\mathbf{S}$, a second set of experiments was designed. First, a fairly large, densely connected initial reconstruction consisting of 404 images, 77864 3D points and involving 236016 variables was obtained. The longest trajectory of image projections included in this

[1] The densities of the eight test sequences are at least 84% and in most cases 100% [5].

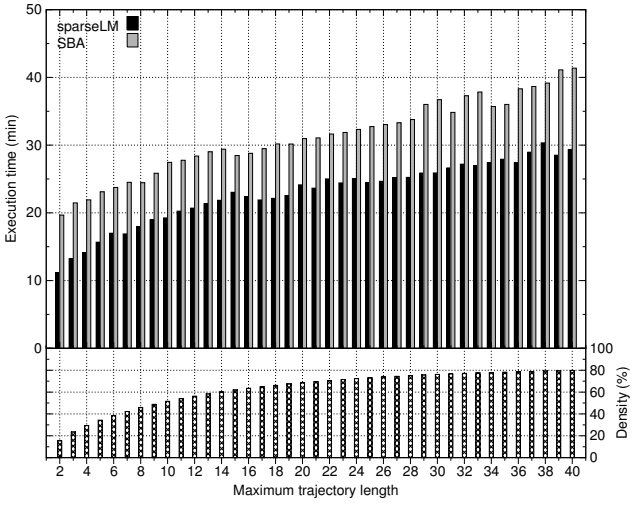
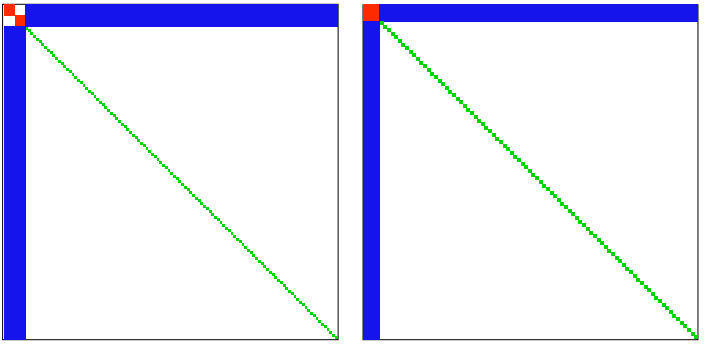


Fig. 2. Performance comparison of Euclidean BA for large reconstructions using `sparseLM` and `sba`: execution time (top) and **S** matrix density (bottom) vs. the maximum trajectory length l

reconstruction has a length of 40. Then, for a length limit l assuming values in $\{2, \ldots 40\}$, several other reconstructions were generated from the initial one by truncating projection trajectories that were more than l images long, taking care to avoid disconnecting the camera network. In this manner, the generated reconstructions differ only in the densities of their point submatrices, thus providing a basis for comparing the performance of sparseLM-based BA against that of `sba` for varying densities of the matrix **S**. The top part of Fig. 2 summarizes the execution times of the two alternatives to BA applied to the 39 generated reconstructions, whereas the bottom part shows their corresponding **S** matrix densities. Clearly, the performance difference between `sparseLM` and `sba` is reversed in favor of the former and is more pronounced for less connected image sets. As has been also observed in [5], this difference stems from the fact that the computation of the dense Cholesky decomposition of **S** has time complexity $O(N^3)$ and thus becomes appreciable for large N ($N = 2424$ in this particular case). Furthermore, the time spent by `sparseLM` for carrying out the the symbolic factorization once in the beginning pays off by enabling it to numerically compute the sparse Cholesky in less time at each subsequent iteration. The downside of using `sparseLM` is that it requires about two to three times more memory than `sba`. This is because direct sparse solvers require additional memory to store the symbolic factorization, whose size depends on the matrix's sparsity structure.

5.2 Trifocal Tensor Estimation

The trifocal tensor $\mathcal{T}$ encapsulates all geometric relations among three images that are independent of scene structure. According to the "Gold Standard"



(a) (b)

Fig. 3. (a) Hessian structure for trifocal tensor estimation involving the 114 3D points visible in the first 3 frames of the "basement" sequence. Color coding is as in Fig. 1. (b) Hessian structure for homography refinement involving 88 image point pairs. Red dots correspond to homography-homography variable pairs, green to point-point pairs and blue to homography-point pairs.

algorithm for obtaining the Maximum Likelihood Estimate of $\mathcal{T}$ [1], p.397 from triplets of corresponding points $\mathbf{x}_i$, $\mathbf{x}_i^{'}$ and $\mathbf{x}_i^{''}$, the procedure proceeds as follows. First, a geometrically valid estimate of $\mathcal{T}$ is computed with a linear algorithm that minimizes the algebraic error and a canonical triplet of camera matrices is recovered from this estimate. Subsidiary variables corresponding to 3D points $\mathbf{X}_i$ are then introduced and initialized via triangulation. $\mathcal{T}$ is parametrized by the elements of the camera matrices $\mathbf{P}^{'}$ and $\mathbf{P}^{''}$. Subsequently, the cost function

$$\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 \; + \; d(\mathbf{x}_i^{'}, \hat{\mathbf{x}}_i^{'})^2 \; + \; d(\mathbf{x}_i^{''}, \hat{\mathbf{x}}_i^{''})^2 \tag{5}$$

is minimized over the 3D points $\mathbf{X}_i$ and the elements of the two camera matrices $\mathbf{P}^{'}$, $\mathbf{P}^{''}$ with $\hat{\mathbf{x}}_i = [\mathbf{I} \mid \mathbf{0}]\mathbf{X}_i$, $\hat{\mathbf{x}}_i^{'} = \mathbf{P}^{'}\mathbf{X}_i$ and $\hat{\mathbf{x}}_i^{''} = \mathbf{P}^{''}\mathbf{X}_i$. For n 3D points, the minimization involves $3n + 24$ variables and amounts to a sparse problem (cf. Fig. 3(a)) solvable by `sparseLM`. Finally, the three correlation slices of $\mathcal{T}$ are set to $\mathbf{T}_i = \mathbf{a}_i\mathbf{b}_4^T - \mathbf{a}_4\mathbf{b}_i^T$, $i = 1 \ldots 3$, where $\mathbf{a}_i$, $\mathbf{b}_i$ are respectively the i-th columns of the refined camera matrices $\mathbf{P}^{'} = [\mathbf{A} \mid \mathbf{a}_4]$, $\mathbf{P}^{''} = [\mathbf{B} \mid \mathbf{b}_4]$. The tensor estimated in this manner satisfies by construction the trilinear constraints for a triplet of refined corresponding points.

The reprojection error of (5) is quite complex and minimizing it involves a large number of parameters. An approximate solution to overcome this is to substitute (5) with the so-called Sampson error [1], p.98, which is the distance to the first order approximation of the algebraic variety defined by the trilinear constraints. Minimization of the sum of Sampson errors for all points relates to only 24 variables, appendix B of [22] provides more details. While it might seem reasonable to expect that the fewer variables of the Sampson approximation will result in faster performance, it is demonstrated next that an application of `sparseLM` performs faster and is more accurate.

Table 2. Statistics for tensor estimation using `sparseLM` (*sLM*), the Sampson approximation (*SA*) and dense LM (*dLM*) approaches. The columns are as follows: Total number of variables for *sLM* & *dLM*, average initial transfer error in pixels, average final transfer error in pixels, total number of objective function/Jacobian evaluations, total number of iterations and linear systems solved, elapsed execution time in seconds. Again, identical values for the user-defined minimization parameters have been used throughout all experiments.

Sequence	# vars	initial	final error		func/jac evals			iter./sys. solved			exec. time		
	sLM & dLM	error	sLM & dLM	SA	sLM	SA	dLM	sLM	SA	dLM	sLM	SA	dLM
"movi"	729	0.330	0.286	0.320	32/30	59/2	38/30	30/32	10/11	30/38	0.42	1.92	43.38
"sagalassos"	681	0.745	0.335	0.737	38/31	80/2	39/34	31/38	31/32	34/39	0.41	2.34	43.52
"arenberg"	960	0.428	0.357	0.428	35/29	41/1	35/31	29/35	16/17	31/35	0.60	1.73	99.83
"basement"	366	0.472	0.397	0.459	35/28	159/4	32/29	28/35	62/63	29/32	0.18	2.53	5.01
"house"	636	0.393	0.367	0.389	60/49	39/1	65/52	49/60	14/15	52/65	0.58	1.07	43.03
"maquette"	1041	0.771	0.429	0.739	37/33	83/2	37/31	33/37	34/35	31/37	0.75	3.81	133.61
"desk"	594	0.545	0.511	0.545	37/30	32/1	34/31	30/37	7/8	31/34	0.32	0.83	23.99
"calgrid"	2097	0.420	0.155	0.320	39/35	62/2	41/34	35/39	13/14	34/41	1.83	5.75	1151.75

The cost function (5) was minimized with `sparseLM` and `levmar` [23], which includes a dense version of the LM algorithm implemented by `sparseLM`. These two approaches are labeled *sLM* and *dLM*, respectively. Furthermore, the total error of the Sampson approximation was minimized with `levmar` using a secant variant of the dense LM; this approach is labeled *SA*. *dLM* serves as a reference for the time savings achieved by *SA* and *sLM*. The three alternative approaches were applied to the estimation of the trifocal tensor corresponding to the first three frames of each sequence used in section 5.1 and the related statistics are presented in Table 2.

The performance of the three approaches is evaluated for accuracy and efficiency, using the average tensorial transfer error for all points in all three frames and the total execution time, respectively. *sLM* and *dLM* employ the same objective function and, therefore, perform identically with respect to accuracy. However, *dLM* is at least two orders of magnitude slower. On the other hand, *SA* is less accurate than *sLM* and, being between 2 to 14 times slower, is also considerably less efficient. The reasons for the worse performance of *SA* can be partly attributed to the fact that the computation of the Sampson error for each point triplet calls for a costly SVD operation to estimate the pseudoinverse of a 9×9 rank 3 matrix [22]. Furthermore, the Jacobian of the Sampson error is too complicated to express analytically, which necessitates its approximation using finite differences that raise the total number of performed SVDs even further. As a matter of fact, the motivation for using a secant variant of dense LM with Broyden's rank one update for minimizing the Sampson error was to ease down the overhead of finite differentiation. The overall superior performance of *sLM* combined with the restrictive assumption made by the Sampson approximation according to which the variety of trilinear constraints has to be well approximated by a first order expansion in the vicinity of the current estimate, clearly suggests *sLM* as the preferred alternative for tensor estimation.

5.3 Homography Estimation

A homography is a general plane to plane projective transformation that is represented by a non-singular homogeneous 3×3 matrix $\mathbf{H}$. Assuming that a set of corresponding coplanar image point pairs $\mathbf{x}_i,\ \mathbf{x}_i^{'}$ is available, the "Gold Standard" algorithm for estimating $\mathbf{H}$ is as follows (cf. [1], p.114): First, an initial estimate is computed using a linear normalized DLT algorithm embedded in a robust regression framework to safeguard against outliers. Then, considering only the inliers, the initial estimate is used as a starting point for minimizing the following geometric cost over $\mathbf{H}$ and the subsidiary points $\hat{\mathbf{x}}_i$:

$$\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 \ + \ d(\mathbf{x}_i^{'}, \mathbf{H}\hat{\mathbf{x}}_i)^2. \tag{6}$$

The minimization corresponds to a sparse problem which involves $9 + 2n$ variables, n being the number of inlying point pairs (cf. Fig. 3(b)). In a manner similar to the estimation of the trifocal tensor, the geometric error of (6) can be approximated with the Sampson error involving 9 variables.

The performance of `sparseLM` minimizing (6) was compared against those of a dense LM algorithm utilized to minimize (6) and the Sampson approximation. Five experiments were carried out using around 900 SIFT keypoints extracted and matched between successive pairs from the six images of the "graffiti" sequence. Although lack of space prevents the inclusion of detailed statistics, it is noted that the performance of `sparseLM` was between 455 to 886 times better than that of the dense LM algorithm minimizing (6) and between only 1.1 to 1.6 times worse than that of the dense LM applied to the Sampson approximation. As expected, minimizing (6) was slightly more accurate than employing the corresponding Sampson approximation.

6 Conclusions

A general-purpose, computationally efficient implementation of sparse non-linear least squares optimization is beneficial to a wide range of vision tasks. This paper has presented an overview of `sparseLM`, a such open source implementation and has demonstrated its versatility and effectiveness in different practical situations. Considering that its applicability extends beyond geometric vision, `sparseLM` can potentially prove invaluable to a variety of research fields and disciplines.

References

1. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
2. Sturm, P., Maybank, S.: A Method for Interactive 3D Reconstruction of Piecewise Planar Objects from Single Images. In: Proc. of BMVC 1999, pp. 265–274 (1999)
3. Sinha, S., Pollefeys, M.: Pan-Tilt-Zoom Camera Calibration and High-Resolution Mosaic Generation. Computer Vision and Image Understanding Journal 103, 170–183 (2006)

4. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle Adjustment – A Modern Synthesis. In: Proc. of the Int'l Workshop on Vision Algorithms: Theory and Practice, pp. 298–372 (1999)
5. Lourakis, M.A., Argyros, A.: SBA: A Software Package for Generic Sparse Bundle Adjustment. ACM Trans. Math. Software 36, 1–30 (2009)
6. Davis, T.: Direct Methods for Sparse Linear Systems. In: Fundamentals of Algorithms. SIAM, Philadelphia (2006)
7. Saad, Y.: Iterative Methods for Sparse Linear Systems. SIAM, Philadelphia (2003)
8. Nocedal, J., Wright, S.: Numerical Optimization. Springer, New York (1999)
9. Gould, N., Scott, J., Hu, Y.: A Numerical Evaluation of Sparse Direct Solvers for the Solution of Large Sparse Symmetric Linear Systems of Equations. ACM Trans. Math. Softw. 33, 10 (2007)
10. Amestoy, P., Davis, T., Duff, I.: Algorithm 837: AMD, an Approximate Minimum Degree Ordering Algorithm. ACM Trans. Math. Softw. 30, 381–388 (2004)
11. Karypis, G., Kumar, V.: A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. SIAM J. Sci. Comput. 20, 359–392 (1998)
12. Barrett, R., Berry, M., Chan, T.F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., der Vorst, H.V.: Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd edn. SIAM, Philadelphia (1994)
13. Davis, T.: Algorithm 849: A Concise Sparse Cholesky Factorization Package. ACM Trans. Math. Softw. 31, 587–591 (2005)
14. Duff, I.: MA57—A Code for the Solution of Sparse Symmetric Definite and Indefinite Systems. ACM Trans. Math. Softw. 30, 118–144 (2004)
15. Schenk, O., Gartner, K.: Solving Unsymmetric Sparse Systems of Linear Equations with PARDISO. J. of Future Generation Computer Systems 20, 475–487 (2004)
16. Demmel, J., Eisenstat, S., Gilbert, J., Li, X., Liu, J.: A Supernodal Approach to Sparse Partial Pivoting. SIAM J. Matrix Analysis and Applications 20, 720–755 (1999)
17. Rotkin, V., Toledo, S.: The Design and Implementation of a New Out-of-Core Sparse Cholesky Factorization Method. ACM Trans. Math. Softw. 30, 19–46 (2004)
18. Davis, T.: Algorithm 832: UMFPACK, An Unsymmetric-Pattern Multifrontal Method. ACM Trans. Math. Softw. 30, 196–199 (2004)
19. Chen, Y., Davis, T., Hager, W., Rajamanickam, S.: Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate. ACM Trans. Math. Softw. 35, 1–14 (2008)
20. Ashcraft, C., Grimes, R.: SPOOLES: An Object-Oriented Sparse Matrix Library. In: Proc. of SIAM Conf. on Parallel Processing for Scientific Computing (1999)
21. Amestoy, P., Duff, I., Koster, J., L'Excellent, J.Y.: A Fully Asynchronous Multifrontal Solver Using Distributed Dynamic Scheduling. SIAM Journal of Matrix Analysis and Applications 23, 15–41 (2001)
22. Torr, P., Zisserman, A.: Robust Parameterization and Computation of the Trifocal Tensor. Image and Vision Computing 15, 591–605 (1997)
23. Lourakis, M.: levmar: Levenberg-Marquardt Nonlinear Least Squares Algorithms in C/C++ (2004), http://www.ics.forth.gr/~lourakis/levmar/

Geometric Image Parsing in Man-Made Environments

Olga Barinova[1,⋆], Victor Lempitsky[2], Elena Tretiak[1], and Pushmeet Kohli[3]

[1] Moscow State University
[2] University of Oxford
[3] Microsoft Research Cambridge

Abstract. We present a new parsing framework for the line-based geometric analysis of a single image coming from a man-made environment. This parsing framework models the scene as a composition of geometric primitives spanning different layers from low level (edges) through mid-level (lines and vanishing points) to high level (the zenith and the horizon). The inference in such a model thus jointly and simultaneously estimates a) the grouping of edges into the straight lines, b) the grouping of lines into parallel families, and c) the positioning of the horizon and the zenith in the image. Such a unified treatment means that the uncertainty information propagates between the layers of the model. This is in contrast to most previous approaches to the same problem, which either ignore the middle levels (lines) all together, or use the bottom-up step-by-step pipeline.

For the evaluation, we consider a publicly available York Urban dataset of "Manhattan" scenes, and also introduce a new, harder dataset of 103 urban outdoor images containing many non-Manhattan scenes. The comparative evaluation for the horizon estimation task demonstrate higher accuracy and robustness attained by our method when compared to the current state-of-the-art approaches.

1 Introduction

Recent years have seen a growing interest in the geometric analysis of a scene based on as little as a single image of this scene. Often the image of interest comes from a man-made environment, e.g. when the image is taken indoors or on a city street. In this case, the image is highly likely to contain a certain number of straight lines, which can be identified in the edgemap of the image, and which often can be further grouped into parallel families. The presence of such lines and their parallelism are known to be valuable cues for the geometric analysis.

When a family of parallel lines is projected on the image, their projections are known to intersect in a single point in the image plane called *vanishing point*. The vanishing point uniquely characterizes the 3D direction of those lines (given

⋆ The first three authors are supported by Microsoft Research programs in Russia. Victor Lempitsky is also supported by EU under ERC grant VisRec no. 228180.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 57–70, 2010.

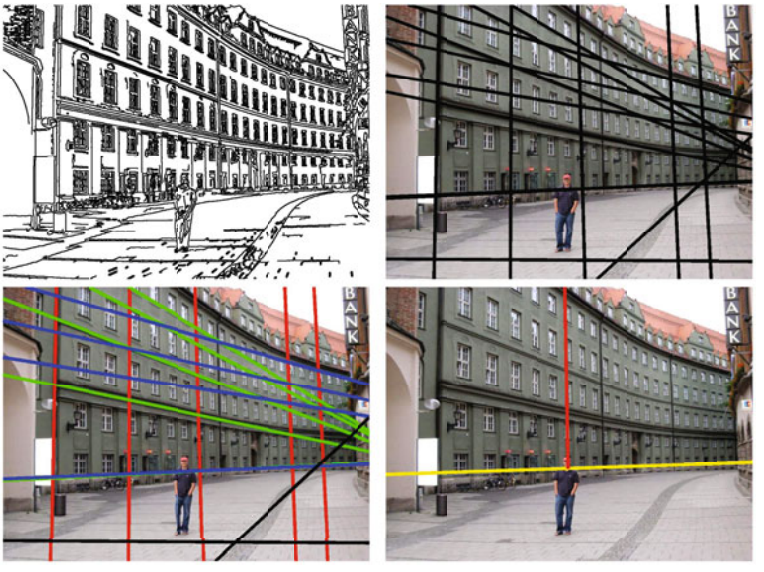


Fig. 1. Geometric primitives of different levels for an example "non-Manhattan" image. *TopLeft* – edge pixels, *TopRight* – straight lines, *BottomLeft* – lines are grouped in parallel families (color indication used), *BottomRight* – the horizon and the zenith (shown with the direction vector in red). Our framework aims at joint estimation of primitives at the latter three levels given the former one (edge pixels).

the camera). When 3D directions of several families are coplanar, the respective vanishing points belong to the same line. Such situation occurs frequently for man-made environments, as there often exist several families with different horizontal directions. In this case, the line containing their vanishing points is called the *horizon*. Most of the remaining lines of the scene are typically vertical. As such, they are parallel to each other and their projections intersect in the vanishing point called the *zenith*[1].

The environments where horizontal lines fall into two orthogonal families, are known as "Manhattan" worlds. A considerable number of previous works investigated the Manhattan case, and the particular simplifications that it brings to the geometric analysis. The parsing framework suggested in this work may be adapted to the Manhattan case, however our work focuses on the non-Manhattan case, assuming the presence of the horizon and the zenith but not the two orthogonal horizontal directions. Surprisingly, very few previous works have paid attention to such scenario (most notably [1]), although we would argue that such assumptions about the scene strike a good balance between the generality and the robustness of the estimation.

[1] Strictly speaking, when this vanishing point lies below the horizon, it should be called the *nadir*. For brevity, we use the term *zenith* in this case as well.

In general, several computer vision and image processing tasks can benefit from the ability to extract the geometric information from a single image. E.g. the knowledge about the location of the horizon may be used to rectify the user photograph with inclined horizon, to facilitate the dense single-view reconstruction and "auto pop-up" [2,3]; this knowledge may also greatly improve semantic segmentation, scene understanding, and object detection [4] as well as video stabilization [5]. The abundance of applications thus motivates the research into better method of geometric analysis of single images leading to more accurate and robust algorithms.

1.1 Related Work

Conceptually, the process of line-based geometric analysis of a single image is well investigated, and typically involves several bottom-up steps. Thus, the process might be initialized with the edge map of an image computed with some edge detector (a standard Canny detector is used in this work). Then, the bottom-up pipeline [6,7,8,9,10,11,12] involves grouping edges into lines, grouping lines into line families and finding the respective vanishing points, and, finally, fitting the horizon and the zenith or the Manhattan directions, depending on the assumptions about the world.

The problem with the step-by-step approach is, however, that neither of the steps can be performed with 100% accuracy and reliability. As the edge maps are always noisy and contaminated with spurious edge pixels not coming from straight lines, the line detection step would miss some of the straight lines and, even worse, detect some spurious lines that do not exist in the scene. Due to these errors, the parallel line grouping step would often group together lines from different families or create groups containing spurious lines (leading to spurious vanishing points) or split actual line families into several (reducing the accuracy of the respective vanishing point estimation). Finally, given an imperfect set of vanishing point, contaminated with outliers, horizon and zenith estimation may lead to gross errors.

Previous works address the challenges associated with each step through several classes of techniques, including robust statistical inference[13], clustering [6,9,14,11], various kinds of Hough transforms [7,9,10], stochastic model fitting [15,12] as well as seeking user supervision [8]. While different approaches possess different strengths and weaknesses, neither results in a perfect accuracy and robustness, leading to the accumulation of errors towards higher stages of the pipeline.

A group of methods [16,17,1,18] goes beyond this pipeline paradigm, as they bypass the line extraction step altogether and directly fit the low-parametric high-level model of the frame (the Manhattan frame [16,17] or a set of Manhattan frames [1]) to the low-level edge map or even to the dense set of image gradients. The joint optimization nature of these methods is similar to our philosophy. However, the simplicity of the model and lack of the edges-to-lines grouping stage limits the accuracy and robustness of their approach as compared to a well-engineered full pipeline approach such as [12].

York Urban dataset [18] The new "Eurasian cities" dataset

Fig. 2. While the York Urban dataset [18] contains images of "Manhattan" worlds, our framework uses less restrictive scene assumptions that are met by non-Manhattan images in the new dataset that we introduce. Our framework is evaluated on both datasets.

1.2 Overview of Our Method

In this work, we investigate the *geometric parsing* approach to the line-based geometric analysis. By geometric parsing here, we understand the process, when the geometric elements at different levels of complexity (Figure 1), as well as the intra-level grouping relations are explicitly recovered through the joint optimization process. Note, that the term *parsing* is used in a similar meaning in such works as [19], where semantic primitives of different levels (e.g. body parts, individual humans, crowd) as well as the intra-level grouping relations are recovered. In our cases, the primitives at different levels are edge pixels, lines, horizontal vanishing points, the zenith and the horizon.

Our work thus differs from works that employ a single bottom-up pass, as the inference in our case is performed jointly, allowing the information from top levels resolve the ambiguities on the lower levels (and vice versa). Our work also differs from the works that bypass the line detection, as the lines in our method are detected explicitly. To the best of our knowledge, the method presented here is the first that integrates line detection, vanishing point location, and higher-level geometric estimation (the horizon and the zenith in our case) in a single optimization framework. Notably, the optimization in our method does not employ alternations between different levels, and is therefore less prone to getting stuck at poor local minima.

There are several design choices and assumptions in our model that are motivated by the applicability and tractability. Firstly, unlike the majority of previous works, we do not make a Manhattan-world assumption. Instead, we consider a less-restrictive non-Manhattan scenario similar to the "Atlanta world" of [1] that will be detailed below in Section 2. Regarding the camera parameters, we assume that the principal point is known (if unknown we assumed it to be in the center of the frame); we also assume that pixels are square. This assumption holds approximately for the vast majority of cameras in real life, and it makes the inference in our model much easier. We also do not model radial distortion explicitly, which is perhaps a bigger shortcoming of our model, although the robust nature of our algorithm means that considerable distortion might be tolerated without explicit modeling. Finally, we assume the focal length unknown.

Theoretically, locations of the horizon and the zenith allow to estimate the focal length of the camera directly from the results of the parsing, however the accuracy of such estimation is hindered by the degeneracy that occurs when the horizon passes near the principal point, which in practice happens very often.

In a sequel, we detail our energy model in Section 2, and discuss the optimization procedure in Section 3. We then perform the experimental validation on two datasets (Figure 2). The first one is the York Urban dataset presented in [18], where several approaches were benchmarked. This dataset has been recently also used for the evaluation in [12], where improved results have been reported. The second dataset was collected by ourselves and, unlike Urban, contains a lot of more challenging non-Manhattan outdoor scenes. The experimental comparison in Section 4 demonstrates the competitiveness of the parsing approach.

2 The Model for Geometric Parsing

We now explain the energy model of the world within our method. We assume an image to be defined by the set of its edge pixels. The main assumptions about the world are a) that a considerable part of edge pixels may be explained by a set of lines, b) that considerable part of those lines fall into several parallel line families. It is further assumed that c) one of these families is a set of vertical (in 3D) lines converging (in the image plane) to the *zenith* and d) all other families consist of horizontal (in 3D) lines converging (in the image plane) to a set of *horizontal* vanishing points, that all lie close to a single line in the image plane known as the *horizon*. The model thus encompasses the edge pixels, the lines, the zenith, and the horizontal vanishing points, as well as the grouping relations of edge pixels in the lines as well as the lines into the parallel families.

We now introduce the notation and the energy model. The edge pixels are denoted $\mathbf{p} = \{p_i\}_{i=1..P}$. The lines present in the scene are denoted $\mathbf{l} = \{l_i\}_{i=1..L}$. As the model involves grouping of lines into parallel families, we denote with z the vanishing point of the vertical line family (the zenith) and with $\mathbf{h} = \{h_i\}_{i=1..H}$ the set of vanishing points of the horizontal families. The points $h_1, h_2 \dots h_H$ thus have to lie close to a line in the image plane (we will refer to this fact as the *horizon constraint*).

The energy function in our method includes the individual energy terms corresponding to the (pseudo-)likelihood of each edge and each line. The edge pixel energy term is defined as:

$$E_{edge}(p|\mathbf{l}) = \min\left(\theta_{bg}, \min_{i=1..L} \theta_{dist} \cdot d(p, l_i) + \theta_{grad} * d_{angle}(p; l_i)\right), \qquad (1)$$

where $d(p, l_i)$ denotes the Euclidean distance in the image plane between the pixel p and the line l_i, $d_{angle}(p; l_i)$ denotes the angular difference between the local edge direction at pixel p and the direction of the line l_i, θ_{bg} is the constant, corresponding to the likelihood of the background clutter, and θ_{dist} and θ_{grad} are the constants corresponding to the spread of edge pixels generated by a particular line around that line. Thus, the energy term for an edge pixel p is

small if this edge pixel is well explained by some line from the set $\mathbf{l}$ and is large otherwise. The largest possible value is θ_{bg}, which corresponds to an edge pixel generated by the background clutter.

The line energy terms are defined as

$$E_{line}(l|\mathbf{h}, z) = \min\left(\eta_{bg}, \min_{i=1..H} \eta_{dist} \cdot \phi(l, h_i)^2, \eta_{dist} \cdot \phi(l, z)^2\right), \tag{2}$$

where ϕ denotes the distance on the Gaussian sphere [20] between the projection of the line l and projection of the respective vanishing point (h_i or z). η_{bg} is the constant, corresponding to the likelihood of lines that are neither horizontal nor vertical, and η_{dist} is the constant, corresponding to the spread of lines in their families around the respective vanishing points. Thus, the energy term for a line l is small if this line is well explained by (i.e. passes close to) a vanishing point from the set $\mathbf{h} \cup \{z\}$ and is large otherwise. The largest possible value is η_{bg}, which corresponds to a line that is neither vertical nor horizontal.

According to horizontal constraint introduced above all vanishing points except the zenith have to lie close to a line in the image plane. How can we enforce this constraint? Should a separate variable for the position of the horizon be introduced? It turns out [21] that under our assumption about internal camera parameters (square pixels and known principal point) this is not necessary. Under these assumptions, the horizon is perpendicular to the radius vector between the line $L(z)$ connecting the zenith and the principal point, and we enforce this perpendicularity with the following energy term:

$$E_{horizon}(u, h|z) = \kappa_{hor} \cdot \tan \psi(u - h, L(z)) \tag{3}$$

where ψ is the absolute angle between the vector $u - h$ and a perpendicular to $L(z)$, and κ_{hor} is a constant. The tan in (3) was chosen because it imposes significant penalty (upto $+\infty$) on strong non-orthogonality between the horizon and $L(z)$.

The final energy is thus defined as:

$$E_{total}(\mathbf{l}, \mathbf{h}, z|\mathbf{p}) = \sum_{i=1..P} E_{edge}(p_i|\mathbf{l}) + \sum_{i=1..L} E_{line}(l_i|\mathbf{h}, z) + \sum_{1 \le i < j \le H} E_{horizon}(h_i, h_j|z) + E_{prior}(\mathbf{l}, \mathbf{h}), \tag{4}$$

where $E_{prior}(\mathbf{l}, \mathbf{h}) = \lambda_{line}|\mathbf{l}| + \lambda_{vp}|\mathbf{h}|$, is an MDL prior penalizing the number of lines $|\mathbf{l}| = L$ and the number of horizontal vanishing points $|\mathbf{h}| = H$, thus favouring simpler explanations of the scene (λ_{line} and λ_{vp} are the constants regulating the strength of this prior). The energy (4) thus ties together the different-level components in the image of a non-Manhattan environment, and the line-based parsing of such an image may be performed through the minimization of (4).

Probabilistic interpretation and the model of [22]. Some of the components of our model may be easily formulated with the language of probabilities. In particular, the part of our model related to the edges and their grouping into

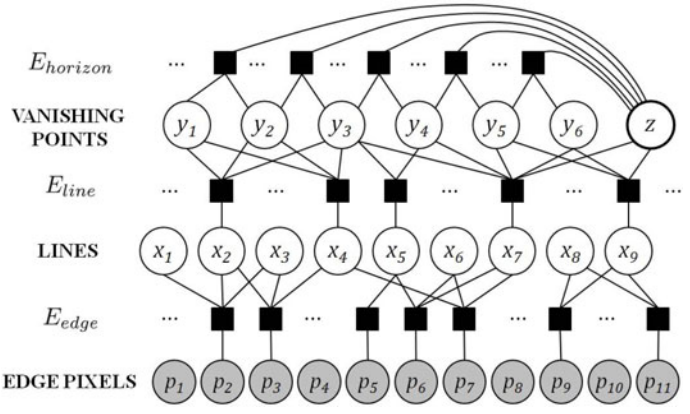


Fig. 3. The graphical model for the discrete approximation of the energy (4). The variables $x_1 \dots x_X$ and $y_1 \dots y_Y$ are binary and correspond to the existence or the absence of candidate lines and horizontal vanishing points. z stands for the location of the zenith and takes the value in a precomputed discrete set of 2D points in the image plane. The unary cliques corresponding to $x_1 \dots x_X$ and $y_1 \dots y_Y$ are omitted for clarity. The shaded nodes (edge pixels) are observed both during training and at test time. *Please, see text for more details.*

lines is in the exact correspondence with the probabilistic model of Hough transform derived in [22]. The next layer of the model concerned with lines and their grouping into families permits an analogous probabilistic treatment. It is unclear, however, if the $E_{horizon}$ term in (3) admits a probabilistic interpretation, as it apparently involves some overcounting of the perpendicularity cues. On practice, this non-probabilistic nature does not present a problem, as we train our model discriminatively by tuning the constants $\theta_{bg}, \theta_{dist}, \theta_{grad}, \eta_{bg}, \eta_{dist}, \kappa_{hor}, \lambda_{line}, \lambda_{vp}$ on the hold-out validation set.

3 Inference

The minimization of (4) is a hard computation problem that necessitates the use of approximations. One possible way would be to minimize it greedily in a layer-by-layer fashion, first choosing the set of lines given the edges, then choosing the set of vanishing points given lines, then fitting the horizon and the zenith into the chosen lines. Such approach would correspond to the traditional bottom-up pipeline from previous methods. Its results might be improved with reiteration of the process through the EM-algorithm, although on practice that suffers from the local minima problem and often gets stuck close to the initial greedy approximation.

A different approach taken in this work is to derive a *discrete approximation* to the original energy that is easier to minimize. To achieve that, we do two steps of the bottom-up pipeline, namely line detection and vanishing point detection,

with very low acceptance thresholds, ensuring that an extensive set of X lines $\hat{l}_1..\hat{l}_X$ and an extensive set of Y vanishing points $\hat{h}_1..\hat{h}_Y$ are detected. On practice, one may use any approach that detects lines based on the edgemap and any approach that detect a set of vanishing points based on lines. We detail our choices in the experimental section (see also Figure 4).

The task of the approximate minimization of (4) may then be reduced to the minimization of the energy of discrete variables $\mathbf{x} = \{x_i\}_{i=1..X}$, $\mathbf{y} = \{y_i\}_{i=1..Y}$, and z. Here, each variable x_i is binary and decides whether a candidate line $\hat{l}_i$ is present ($x_i = 1$) or absent ($x_i = 0$) in the image. Similarly, each variable y_i is binary and decides whether a candidate vanishing point $\hat{h}_i$ is a *horizontal* vanishing point that is present ($y_i = 1$) or absent ($y_i = 0$) in the image. Finally, the variable z is, as defined above, a 2D point in the image plane corresponding to the zenith. The set of its possible locations is however restricted to discrete set of candidate vanishing points. For computational efficiency, we may further prune the set of possible locations for z by removing candidate vanishing points that correspond to the horizon inclinations of more than 7.5 degrees. This can be regarded as an additional hard prior on z in our original energy.

The discrete approximation to the energy (4) is then defined by the requirement:

$$E_{discrete}(\mathbf{x}, \mathbf{y}, z|\mathbf{p}) \equiv E_{total}(\{\hat{l}_j\}_{j:x_j=1}, \{\hat{h}_k\}_{k:y_k=1}, z|\mathbf{p})\,. \tag{5}$$

In other words, the discrete energy is defined as the continuos energy of the appropriate subsets of candidate lines and vanishing points.

In more detail, the discrete energy defined in (5) can be written as:

$$E_{discrete}(\mathbf{x}, \mathbf{y}, z|\mathbf{p}) = \sum_{i=1..P} E_{edge}(p_i|\{\hat{l}_j\}_{j:x_j=1}) + \sum_{i=1..X} x_i \cdot E_{line}(\hat{l}_i|\{\hat{h}_k\}_{k:y_k=1}, z) + \sum_{1\le i<j\le Y} y_i \cdot y_j \cdot E_{horizon}(\hat{h}_i, \hat{h}_j|z) + \sum_{i=1..X} \lambda_{line} \cdot x_i + \sum_{i=1..Y} \lambda_{vp} \cdot y_i. \tag{6}$$

The factor graph for the formula (6) is shown in Figure 3. Note, that due to the truncation effect of the constants θ_{bg} and θ_{dist} in the definition of E_{edge} and E_{line}, the connections between the E_{edge} factors and the line variables as well as between the E_{line} factors and the vanishing points variables are sparse. Each E_{edge} factor is connected only to the lines that pass nearby that edge pixel and, likewise, each E_{line} factor is connected to the vanishing point variables that lie near that line.

Since the values of $\mathbf{p}$ are observed, very big efficiency gains may be easily obtained by merging (summing up) the E_{edge} factors that are connected to the same (or nested) sets of line variables. Since E_{edge} terms constitute the vast majority of terms in (6), this trick dramatically reduces the number of energy terms in the model. It permits us to use quite a simple and brute-force optimization scheme, while still allowing short optimization runtime of several seconds for a typical photograph. In more detail, we exhaustively search through the zenith candidate set (which typically includes less then a dozen of candidates). Given a fixed z, we then perform optimization over the binary variables x and

y through the Iterated Conditional Modes algorithm [23] with the randomized node visiting order.

4 Experiments

Technical details. In our experiments we used the following strategy for choosing candidate lines and candidate vanishing points. For the line detection the probabilistic version of Hough transform [22] was used, where we set the parameters of the method to θ_{bg} and θ_{dist} accordingly. As [22] provides the confidence measure for each detected line, we fixed the number of candidates to 500 and for each image took 500 lines with the highest confidence. Figure 4 gives a typical example of what the candidate set typically looks like.

The candidates for vanishing points were chosen using the J-linkage procedure, described in [12]. This method is based on random sampling, so we ran it several times starting from different random initializations. Usually we got from 50 to 100 candidates for vanishing points. After performing the inference in our model we usually got from 2 to 5 vanishing points and groups of lines supporting each of them.

In the experiments on York Urban dataset we exploited the coordinates of principal point provided, in the experiments on the new dataset we assumed the principal point to lie in the center of the image frame.

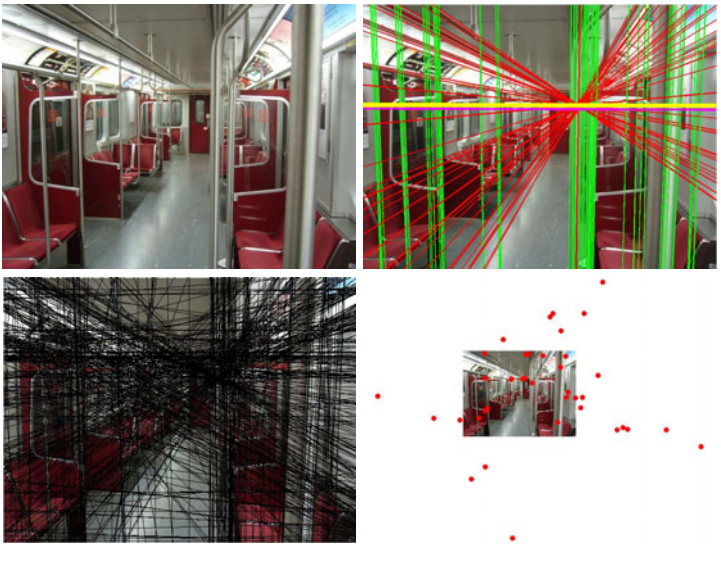


Fig. 4. Sample image from the York Urban dataset: *TopLeft* – the input, *BottomLeft* – all candidate lines superimposed, *BottomRight* – all candidate vanishing points, *TopRight* – the result of the parsing. Coloring reflects grouping into parallel families. Yellow and pink thick lines correspond to the found and the ground truth horizons respectively (the pink line is mostly occluded due to a good fit between the two).

Datasets. Our approach is evaluated on two datasets (Figure 2):

1. The *York Urban* dataset [18] contains 102 images of outdoor and indoor scenes taken within the same location with the same camera. Most of the scenes meet the Manhattan world assumption, as the lines available in the scene mostly fall into the three orthogonal families.

2. The *Eurasian cities* dataset is a new set of 103 outdoor urban images. The images come from the cities of different cultures, hence with different line statistics. They were also taken with different cameras. The main difference of the dataset is the abundance of scenes that fit poorly to the Manhattan assumption. During the annotation, we manually specified several most distinctive lines per each distinctive parallel line family in each image (with the interactive tool similar to that of [18]). This allows to estimate the horizon with good accuracy and we use it as ground truth in the comparative evaluation.

Competing methods. We have compared our approach against the two previously published methods:

1. The method of Tardif [12] is a pipeline approach which reported the top performance on the York Urban dataset. For the experiments on the York Urban dataset we used the author code (with the exception of the EM process that was not published and that we reimplemented by carefully following the text of [12]). For York Urban dataset in cases where more than 3 vanishing points were detected, we chose 3 most orthogonal of them as described in the paper [12]. The coordinates of principal point provided by the authors of the dataset were used during orthogonalization. For the experiments in the Eurasian Cities we did not choose most orthogonal points because the dataset contains non-Manhattan scenes. Parameters of EM were chosen on validation set.

2. The method of Kosecka and Zhang [14] is an approach based on the EM-algorithm, alternating between the two stages: estimation of vanishing point coordinates given distribution of corresponding line segments and re-estimation of distribution of line segments according to positions of vanishing points. The process starts with clustering line segments according to their orientation which results in excessive number of clusters. During EM the clusters with close vanishing points are merged together. Also clusters that have little support are pruned. We took the code from implementation of Automatic Photo Pop Up system [3], which uses that method for vanishing points estimation. Parameters of the method were tuned on the validation sets.

Importantly, to put all the methods on an equal footing, we made sure that all three algorithms are provided with the same Canny edge map (we used the parameters suggested by Tardif in [12]). Both baseline methods use line segments, so we use the line segments detection implementation by [12] for both of them.

After running each method we obtain the zenith, as well as a number of vanishing points corresponding to the parallel families of the line segments (for baseline methods) or lines (for our method). We use this information to estimate the position of the horizon in an image. The horizon is estimated in the same way for all methods. Thus, we restrict it to be perpendicular to the line connecting principal point and zenith. So the slope of horizon is given by zenith and we

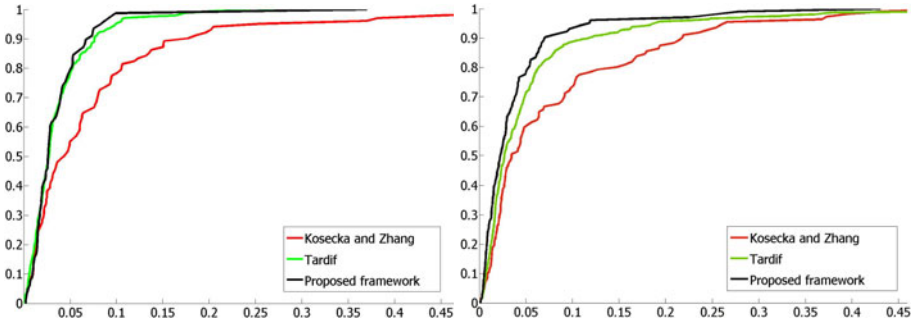


Fig. 5. The results of the comparison of the cumulative statistics for the accuracies of the proposed framework along with the methods of [12] and [14]. The x-axis corresponds to the horizon estimation error measure (see text for more details). The **y**-axis corresponds to the share of the images in the test set that has the error less than the respective x value. In both cases, the proposed framework obtains higher accuracy than the competitors.

estimate only its position along the 1D axis. To do this last step, we perform the weighted least squares fit, where the weight of each detected horizontal vanishing point equals the number of corresponding lines (or line segments).

Accuracy measure. While all the considered approaches essentially output both low-level and high-level primitives, comparing the accuracy of the low-level description of the scenes (e.g. set of lines) is problematic, as the ground truth available for the datasets do not provide full set of lines. Thus, if a line or a vanishing point is present in the output that is missing in the ground truth, it is unclear whether this is due to the error of the algorithm or due to the incompleteness of the ground truth.

We therefore focused on the accuracy of the horizon estimation. Assume that the horizon is given as a (linear) function $H(x)$ of a pixel x-coordinate. Assume that $H_0(x)$ and $H_1(x)$ are the ground truth and the estimated horizon. We then define the estimation error as the maximum of $|H_0(x) - H_1(x)|$ over the image domain ($0 < x <$ image width), divided by the image height. To represent the error over the dataset, we plot the share of the images with the error less then τ for each τ.

Results. Quantitative results are given in Figure 5, while in Figure 4 and Figure 6, we present some qualitative examples from both datasets for our framework. Note that we used the first 25 images of each dataset as a held-out set for the parameter validation for all three competing methods[2]. During the

[2] Through the validation, the parameters for our method for York/Eurasian cities were set to: $\theta_{bg} = 8{\cdot}10^{-5}/7.6{\cdot}10^{-5}, \theta_{dist} = 4{\cdot}10^{-5}/3{\cdot}10^{-5}, \theta_{grad} = 4{\cdot}10^{-5}/2{\cdot}10^{-5}, \eta_{bg} = 0.1/0.1, \eta_{dist} = 1.0/0.8, \lambda_{vp} = 0.015/0.015, \lambda_{line} = 0.003/0.01, \kappa_{hor} = 2.0/5.0$. All angular differences were measured in radians.

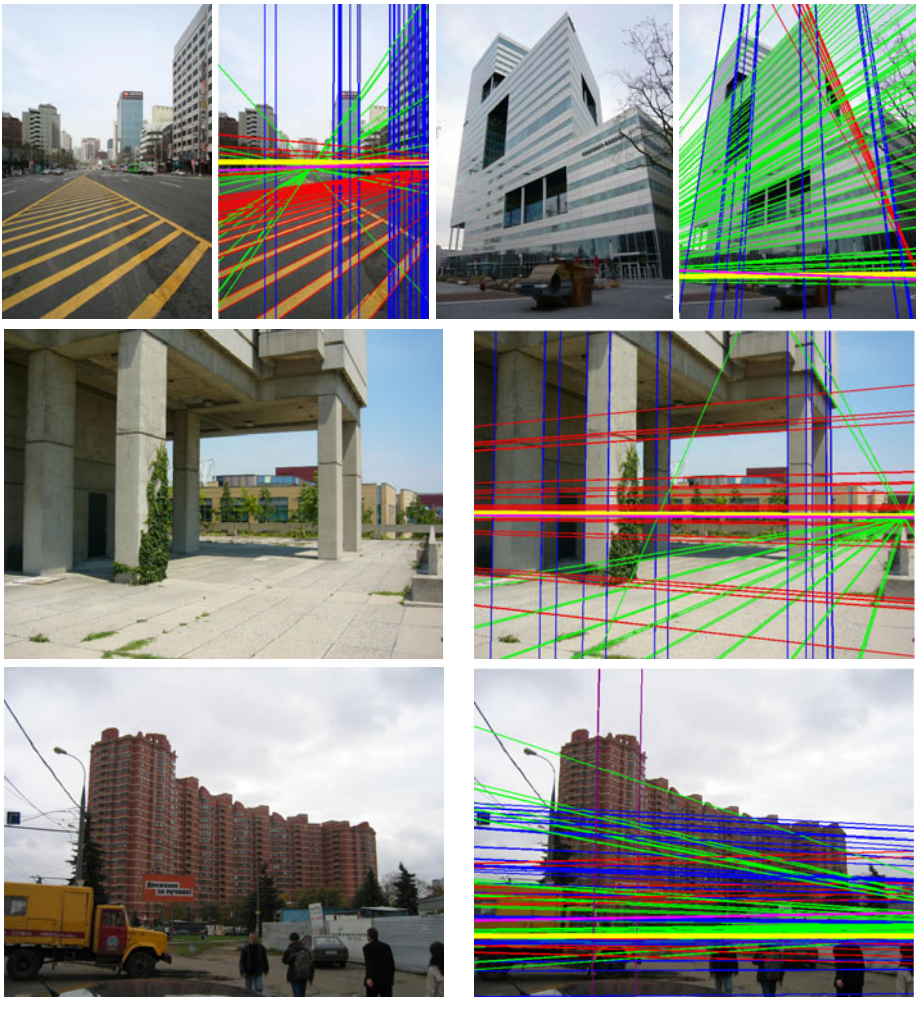


Fig. 6. Sample results of the proposed framework from both datasets. In each pair, we give the input image and the output of the parsing. Coloring reflects grouping into parallel families. Yellow and pink thick lines correspond to the found and the ground truth horizons respectively. Top rows show examples of successful applications, while the bottom one demonstrates one of the worse cases (due to the severely cluttered edge map, the horizon has been estimated significantly below the ground truth).

validation, the area under curve statistics on the hold-out set was optimized. The accuracy measures in the plots in Figure 5 thus reflect the performance on the rest of the images.

As can be seen, the method presented in the paper outperforms both competing methods considerably on the Eurasian cities dataset and performs on a par with [12] and much better than [14] on the York Urban dataset. The latter

is all the more important, given the fact the stronger competing method [12] makes explicit use of the Manhattan assumption that is very appropriate for the York dataset, while our method worked with the more general non-Manhattan world model. At the same time, our current implementation is much slower than the competing methods (few minutes per image vs. few seconds per image on a modern PC). The time for our method is dominated by the candidate (lines and VPs) generation and graph construction, and can be reduced significantly is less exhaustive number of candidates would be considered.

In addition to our main error measure (horizon accuracy), we also estimated the error of the zenith estimation on York urban dataset (where ground truth Manhattan geometry allows accurate localization of the zenith). We measured the errors as the angle between directions to the ground truth zenith and the estimated zenith on a Gaussian sphere [20]. The error for our method (0.0118 ± 0.0292) and for the method [14] (0.0133 ± 0.0139) were lower than the error-rate for [12] (0.0402 ± 0.1918).

5 Summary and Discussion

We formulated the problem of geometric analysis of a single image in an optimization framework. Given a set of observed edge pixels, the framework jointly infers groupings of edge pixels into lines, parallel lines, vanishing points and geometric concepts such as the zenith and the horizon. This framework has advantages over previous bottom up methods for inference of such geometric properties; the most significant one being the ability to incorporate a confidence measure about scene elements in a joint framework.

We observed that many failures of the algorithms resulted from the clutter in the edge map (Figure 6 gives an example). As demonstrated by previous works (e.g. [12]), the effect of the clutter may be reduced substantially by local grouping into line segments. In our framework, this can be accomplished by augmenting the graphical model with one more layer situated between the edge pixels layer and the lines layer.

The current framework also ignores appearance information from the scene elements. For instance, parallel lines arising due to a railway track or a road might have similar appearance which may provide additional cues for grouping lines and inferring the location of the zenith and the horizon. This information can produce better results and is a topic for future work. Another interesting direction of work is the incorporation of an uncertainty measure in the presence of edges.

References

1. Schindler, G., Dellaert, F.: Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In: CVPR, vol. (1), pp. 203–209 (2004)

2. Hoiem, D., Efros, A.A., Hebert, M.: Geometric context from a single image. In: ICCV, pp. 654–661 (2005)
3. Hoiem, D., Efros, A.A., Hebert, M.: Automatic photo pop-up. ACM Trans. Graph. 24, 577–584 (2005)
4. Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. International Journal of Computer Vision 80, 3–15 (2008)
5. Duric, Z., Rosenfeld, A.: Image sequence stabilization in real time. Real-Time Imaging 2, 271–284 (1996)
6. McLean, G.F., Kotturi, D.: Vanishing point detection by line clustering. IEEE Trans. Pattern Anal. Mach. Intell. 17, 1090–1095 (1995)
7. Tuytelaars, T., Gool, L.J.V., Proesmans, M., Moons, T.: A cascaded hough transform as an aid in aerial image interpretation. In: ICCV, pp. 67–72 (1998)
8. Cipolla, R., Drummond, T., Robertson, D.P.: Camera calibration from vanishing points in image of architectural scenes. In: BMVC (1999)
9. Antone, M.E., Teller, S.J.: Automatic recovery of relative camera rotations for urban scenes. In: CVPR, pp. 2282–2289 (2000)
10. Almansa, A., Desolneux, A., Vamech, S.: Vanishing point detection without any a priori information. IEEE Trans. Pattern Anal. Mach. Intell. 25, 502–507 (2003)
11. Aguilera, D.G., Lahoz, J.G., Codes, J.F.: A new method for vanishing points detection in 3d reconstruction from a single view. In: Proc. of ISPRS Commission V (2005)
12. Tardif, J.P.: Non-iterative approach for fast and accurate vanishing point detection. In: ICCV (2009)
13. Collins, R., Weiss, R.: Vanishing point calculation as a statistical inference on the unit sphere. In: ICCV, pp. 400–403 (1990)
14. Kosecká, J., Zhang, W.: Video compass. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 476–490. Springer, Heidelberg (2002)
15. Rother, C.: A new approach for vanishing point detection in architectural environments. In: BMVC (2000)
16. Coughlan, J.M., Yuille, A.L.: Manhattan world: Compass direction from a single image by bayesian inference. In: ICCV, pp. 941–947 (1999)
17. Deutscher, J., Isard, M., MacCormick, J.: Automatic camera calibration from a single manhattan image. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 175–205. Springer, Heidelberg (2002)
18. Denis, P., Elder, J.H., Estrada, F.J.: Efficient edge-based methods for estimating manhattan frames in urban imagery. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 197–210. Springer, Heidelberg (2002)
19. Tu, Z., Chen, X., Yuille, A.L., Zhu, S.C.: Image parsing: Unifying segmentation, detection, and recognition. International Journal of Computer Vision 63, 113–140 (2005)
20. Barnard, S.: Interpreting perspective images. Artificial Intelligence 21, 435–462 (1983)
21. Beardsley, P., Murray, D.: Camera calibration using vanishing points. In: BMVC, pp. 416–425 (1992)
22. Barinova, O., Lempitsky, V., Kohli, P.: On detection of multiple object instances using hough transforms. In: CVPR (2010)
23. Besag, J.: On the statistical analysis of dirty pictures. Journal of the Royal Statistical Society B-48, 259–302 (1986)

Euclidean Structure Recovery from Motion in Perspective Image Sequences via Hankel Rank Minimization

Mustafa Ayazoglu, Mario Sznaier, and Octavia Camps*

Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115, USA

Abstract. In this paper we consider the problem of recovering 3D Euclidean structure from multi-frame point correspondence data in image sequences under perspective projection. Existing approaches rely either only on geometrical constraints reflecting the rigid nature of the object, or exploit temporal information by recasting the problem into a nonlinear filtering form. In contrast, here we introduce a new constraint that implicitly exploits the *temporal ordering* of the frames, leading to a provably correct algorithm to find Euclidean structure (up to a single scaling factor) without the need to alternate between projective depth and motion estimation, estimate the Fundamental matrices or assume a camera motion model. Finally, the proposed approach does not require an accurate calibration of the camera. The accuracy of the algorithm is illustrated using several examples involving both synthetic and real data.

Keywords: Structure from Motion, Perspective Images, Rank Minimization.

1 Introduction

Recovering 3D structure from a sequence of 2D images has been the subject of substantial research [1,2]. For the orthographic projection case, Tomasi and Kanade [3] proposed a method based on factorizing a matrix containing the coordinates of the tracked points, which is forced to have at most rank 4. The method has been extended to paraperspective [4,5] and perspective [6,7] projection. In the former case, the algorithm relies on the estimation of a set of point–dependent *projective depths*. Sturm and Triggs [6] proposed to recover these depths by using the epipolar constraint between two views, which in turn requires estimating the fundamental matrix. Triggs [7] extended this method by refining the projective depths through an iterative procedure alternating with factorization. Other iterative approaches include [8,9,10,11].

Often, factorization techniques are followed by a bundle adjustment to minimize the 2D re-projection error [12,13,14,15,16,17]. In general, this entails a non-linear optimization based on descend methods which are very sensitive to initialization. [9] avoids

* This work was supported in part by NSF grants IIS–0713003 and ECCS–0901433, AFOSR grant FA9550–09–1–0253, and the Alert DHS Center of Excellence under Award Number 2008-ST-061-ED0001.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 71–84, 2010.

this problem by solving a sequence of eigenvalue problems, but convergence cannot be guaranteed.

A common feature of the approaches described above is the fact that they rely entirely on geometrical constraints, discarding temporal information[1]. Indeed, most of these methods are based on quasi-linear algorithms that alternate between estimating the structure and projections, and whose convergence cannot be guaranteed [18,19,20], and the resulting solutions are invariant with respect to frame permutations.

Temporal correlations have been exploited to solve the related problem of simultaneous localization and estimation (SLAM), where the goal is to use data provided by a single moving platform to reconstruct its 3 D trajectory and a local map. In this context, temporal information is exploited by recasting the problem as a non-linear filtering one. The goal is to estimate a state vector that contains the motion state of the moving sensor (e.g. position, velocity, pose) and the 3D coordinates of given features, as well as a probability density function that quantifies the uncertainty in this estimation. Earlier approaches to SLAM required the use of additional sensor data, e.g. odometry or stereo, while later ones, [21] avoid this by requiring a short calibration run using a landmark with a known position. In principle, success of this approach hinges upon the availability of a motion model for the camera, and access to the inputs to the model. While this additional information is typically available in robotic applications, this is not the case for sequences generated by an unknown camera (or object) motion. This difficulty can be circumvented by assuming a simple model (e.g. constant velocity or acceleration [21]), subject to uncertainty. However this leads to larger uncertainty in the estimated feature position. Alternatively, [22] avoid this issue by using the dynamics for tracking only, while reconstructing the 3-D geometry by first triangulating two key-frames obtained during an initialization stage with user input, followed by epipolar search when new keyframes are added and local bundle adjustment. While SLAM methods work well in practice, convergence to the true depths cannot be guaranteed due to uncertainty in the motion model, coupled with the non-convex nature of bundle adjustment. Further, (external) calibration data is usually unavailable in pure SfM applications.

In this paper, we present a convex-optimization based solution to the problem of Euclidean 3D structure recovery from an image sequence under perspective projection. The proposed method avoids the estimation of epipolar geometry and the fundamental matrix. This is accomplished by exploiting the temporal information encoded in the ordering of the given image sequence to recast the problem into a rank minimization form, that can be efficiently solved using existing convex relaxations. The main theoretical result of the paper shows that indeed the solution to this rank-minimization problem recovers the correct Euclidean depths of the scene points, up to a *single* constant scaling factor for all points across the entire motion sequence. This result is general, and neither depends on the object motion model nor necessitates explicitly finding it. The effectiveness of the algorithm is illustrated with several examples involving both synthetic and real data with known ground truth.

[1] In general, the temporal ordering of the frames is *only* used while tracking the features and establishing correspondences across frames.

2 3D Structure from Perspective Images

Consider a camera Cartesian coordinate system defined with its origin at the center of projection and its Z axis along the camera optical axis. Let N be the number of points of a moving rigid object, and let $\mathbf{P}_{ij} = (X_{ij}, Y_{ij}, Z_{ij})^T$ be the 3D Cartesian camera coordinates of point $\mathbf{P}_j$, $j = 1, \ldots, N$, at time i, $i = 1, \ldots, F$. Then, the corresponding 2D image coordinates at time i, $\mathbf{p}_{ij}(u_{ij}, v_{ij})$, are given by

$$u_{ij} = f\frac{X_{ij}}{Z_{ij}} - c_u,\ v_{ij} = \alpha f\frac{Y_{ij}}{Z_{ij}} - c_v \tag{1}$$

where f is the camera's focal length, α is its pixel aspect ratio and (c_u, c_v) is its principal point. In the sequel, for notational simplicity we will assume that $(c_u, c_v) = (0, 0)$[2]. With this notation, the problem of interest here can be formalized as follows.

Problem 1: Given the above setup, find the 3D scene structure $\mathbf{P}_{ij}$ from the $N \times F$ feature correspondences $\mathbf{p}_{ij}$.

Classically, this problem has been solved using the Strum Triggs Algorithm [6], based on iteratively computing the best rank 4 approximation to a matrix constructed from the image data, and the associated projective depths. Since the problem is not jointly convex, this algorithm is guaranteed to converge only to a local solution. Further, the algorithm as stated above can only recover the 3D structure up to an arbitrary (time–varying) projectivity. Recovering the Euclidian geometry entails an additional computationally challenging non–linear, non–convex optimization.

3 Preliminaries

Below we introduce some preliminary definitions required to recast Problem 1 as a rank minimization problem.

Definition 1. *An operator $\mathcal{L}$ that maps a vector $x_o \in R^n$ to an infinite sequence of vectors $x_k \doteq \{\mathcal{L}[x_o]\}_k \in R^n$ is said to be* point-wise rigid *if*

$$\|\{\mathcal{L}\left[\mathbf{P} - \mathbf{Q}\right]\}_k\|_2 = \|\mathbf{P} - \mathbf{Q}\|_2 \text{ for all } \mathbf{P}, \mathbf{Q}, k$$

Definition 2. *N points $\mathbf{P}_1, \ldots, \mathbf{P}_N \in R^3$ are said to belong to a rigid body if, for each frame k, there exist a point $\mathbf{O}_k \in R^3$ (not necessarily in the object) and a point-wise rigid operator $\mathcal{L}$ such that for all points and all time instants, the corresponding trajectories satisfy: $\mathbf{P}_{ki} - \mathbf{O}_k = \{\mathcal{L}\left[\mathbf{P}_{oi} - \mathbf{O}_o\right]\}_k$, $k = 1, 2, \ldots$ where $\mathbf{P}_{ki}$ denote the coordinates of point $\mathbf{P}_i$ at time k.*

For example, for a constant rotation R about a moving axis we have $\{\mathcal{L}\left[\mathbf{P}_{oi} - \mathbf{O}_o\right]\}_k = R^k\left[\mathbf{P}_{oi} - \mathbf{O}_o\right]$.

Definition 3. *Given a vector sequence $\{\mathbf{y}_k\}_{k=1}^{n+l-1}$ its Hankel matrix is defined as:*

$$\mathbf{H}_{\mathbf{y},n,l} \doteq \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_l \\ \mathbf{y}_2 & \mathbf{y}_3 & \cdots & \mathbf{y}_{l+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_n & \mathbf{y}_{n+1} & \cdots & \mathbf{y}_{l+n-1} \end{bmatrix}$$

[2] By redefining, if necessary, $\hat{u}_{ij} = u_{ij} + c_u$ and $\hat{v}_{ij} = v_{ij} + c_v$.

4 Recovering Geometry from Hankel Rank Minimization

In this section, we show that the Euclidean structure of a rigid object undergoing a point-wise rigid transformation can be recovered (up to a single scaling factor) by minimizing the rank of the Hankel matrix associated with the trajectory, subject to one linear and two rank constraints. From its definition, it is clear that the rank of the Hankel matrix encapsulates temporal correlations, since it is not invariant under a permutation of the ordering of the frames. The surprising result is that this rank also encapsulates rigidity, since as we prove below, the correct 3D rigid geometry, up to an overall constant scaling factor, is precisely the one that minimizes it, subject to the additional constraints. This result allows for recasting Problem 1 into a rank-minimization form.

Theorem 1. *Consider the image trajectories* $\mathbf{p}_{ki} = (u_{ki}, v_{ki})^T$, $i = 1,2,3$, $k = 1,\ldots,F$ *of the perspective projections of three points* $\mathbf{P}_{ki}$, $i = 1,2,3$, *belonging to a rigid moving under some point-wise rigid motion operator* $\mathcal{L}$. *Then, the 3D camera Cartesian coordinates of* $\mathbf{P}_{ki}$ $i = 1,2,3$, $k = 1,\ldots,F$ *are given by:*

$$\mathbf{P}_{ki} = \begin{bmatrix} X_{ki} \\ Y_{ki} \\ Z_{ki} \end{bmatrix} = \frac{1}{\lambda_o \rho^k} Z^*_{ki} \begin{bmatrix} \frac{1}{f} u_{ki} \\ \frac{1}{\alpha f} v_{ki} \\ 1 \end{bmatrix} \tag{2}$$

where λ_o *and* $\rho > 0$ *are constant factors (point and frame independent), and where* $\{Z^*_{k1}, Z^*_{k2}, Z^*_{k3}\}_{k=1,\ldots,F}$ *solve the following rank minimization problem*

$$\min_{\{Z^*_{k1}, Z^*_{k2}, Z^*_{k3}\}_{k=1,\ldots,F}} \operatorname{rank}\left(\begin{bmatrix} \mathbf{H}_{\mathbf{y}^{13}} & \mathbf{H}_{\mathbf{y}^{23}} \end{bmatrix}\right) \text{ subject to: } Z_{ki} \geq 1 \tag{3}$$

where

$$\mathbf{y}^{ij}_k = \begin{bmatrix} \frac{1}{f}(Z^*_{ki} u_{ki} - Z^*_{kj} u_{kj}) \\ \frac{1}{\alpha f}(Z^*_{ki} v_{ki} - Z^*_{kj} v_{kj}) \\ Z^*_{ki} - Z^*_{kj} \end{bmatrix}$$

and $\mathbf{H}_{\mathbf{y}} \doteq \mathbf{H}_{\mathbf{y},\lfloor F/2 \rfloor, F}$, *the Hankel matrix of the sequence* $\{\mathbf{y_k}\}_{k=1}^{F}$.

Proof: See the Appendix.

Theorem 1 allows for recovering the correct *relative* 3D structure by solving a rank–minimization problem. This follows from the fact that since $Z^*_{ki} = \lambda_o \rho^k Z_{ki}$, then $\frac{Z^*_{ki}}{Z_{ki}} = \frac{Z^*_{kj}}{Z_{kj}}$ for all (i,j), where Z and Z^* denote the actual and recovered depths, respectively. While in many situations this may suffice, in others it is of interest to recover the geometry up to an overall, frame-independent scaling. As we show next, this can be accomplished by adding one linear and two rank constraints to the problem.

Corollary 1. *The correct 3D geometry (up to a single constant scaling factor) satisfies (3), subject to one linear and two rank constraints.*

Proof. Note that the solutions to (3) satisfy: $\|\mathbf{P}_{ki} - \mathbf{P}_{kj}\|_2^2 = \left(\frac{1}{\lambda_o \rho^k}\right)^2 \|\mathbf{P}^*_{ki} - \mathbf{P}^*_{kj}\|_2^2$ where $\mathbf{P}^*_{ki} \doteq Z^*_{ki} \left[\frac{u_{ki}}{f} \; \frac{v_{ki}}{\alpha f} \; 1\right]^T$. Next, impose rigidity of the reconstructed trajectories *only* across the first and last frames, leading to:

$$\begin{aligned} 0 &= \|\mathbf{P}^*_{Fi} - \mathbf{P}^*_{Fj}\|_2^2 - \|\mathbf{P}^*_{1i} - \mathbf{P}^*_{1j}\|_2^2 \Rightarrow \\ 0 &= \left(\lambda_o \rho^F\right)^2 \|\mathbf{P}_{Fi} - \mathbf{P}_{Fj}\|_2^2 - (\lambda_o \rho)^2 \|\mathbf{P}_{1i} - \mathbf{P}_{1j}\|_2^2 \Rightarrow \rho = 1 \end{aligned} \tag{4}$$

where the last equality follows from the fact that the actual trajectories satisfy $\|\mathbf{P}_{ki} - \mathbf{P}_{kj}\|_2 = \text{constant}$, for all k. Thus, imposing rigidity of the reconstructed object *only* for 2 points across the first and last frames forces the overall scaling to become frame independent (e.g. $\alpha_k = \lambda_o(1)^k = \lambda_o$). As we show below, the constraint (4) can be recast as a combination of linear and rank constraints. Start by rewriting the constraint $\|\|\mathbf{P}^*_{11} - \mathbf{P}^*_{12}\|_2^2 = \|\|\mathbf{P}^*_{F1} - \mathbf{P}^*_{F2}\|_2^2$ as:

$$Z_{11}^2(\frac{u_{11}^2}{f^2} + \frac{v_{11}^2}{f^2\alpha^2} + 1) + Z_{12}^2(\frac{u_{12}^2}{f^2} + \frac{v_{12}^2}{f^2\alpha^2} + 1) - 2 * Z_{11}Z_{22}(\frac{u_{11}u_{12}}{f^2} + \frac{v_{11}v_{12}}{f^2\alpha^2} + 1) -$$
$$Z_{F1}^2(\frac{u_{F1}^2}{f^2} + \frac{v_{F1}^2}{f^2\alpha^2} + 1) - Z_{F2}^2(\frac{u_{F2}^2}{f^2} + \frac{v_{F2}^2}{f^2\alpha^2} + 1) + 2 * Z_{F1}Z_{F2}(\frac{u_{F1}u_{F2}}{f^2} + \frac{v_{F1}v_{F2}}{f^2\alpha^2} + 1) = 0 \tag{5}$$

Next, define the following variables:

$$m_t^{20} \doteq Z_{t1}^2,\ m_t^{11} \doteq Z_{t1}Z_{t2},\ m_t^{02} \doteq Z_{t2}^2 \tag{6}$$

In terms of these new variables, (5) can be rewritten as the *linear* constraint:

$$m_1^{20}(\frac{u_{11}^2}{f^2} + \frac{v_{11}^2}{f^2\alpha^2} + 1) + m_1^{02}(\frac{u_{12}^2}{f^2} + \frac{v_{12}^2}{f^2\alpha^2} + 1) - 2 * m_1^{11}(\frac{u_{11}u_{12}}{f^2} + \frac{v_{11}v_{12}}{f^2\alpha^2} + 1) -$$
$$m_F^{20}(\frac{u_{F1}^2}{f^2} + \frac{v_{F1}^2}{f^2\alpha^2} + 1) - m_F^{02}(\frac{u_{F2}^2}{f^2} + \frac{v_{F2}^2}{f^2\alpha^2} + 1) + 2 * m_F^{11}(\frac{u_{F1}u_{F2}}{f^2} + \frac{v_{F1}v_{F2}}{f^2\alpha^2} + 1) = 0 \tag{7}$$

Further, it can be easily seen[3] that (6) is equivalent to

$$\text{rank}\left\{\begin{bmatrix} m_t^{20} & m_t^{11} \\ m_t^{11} & m_t^{02} \end{bmatrix}\right\} = 1,\ t = \{1, F\} \tag{8}$$

□

From this corollary, it follows that the 3D geometry (up to a single scaling factor) of a moving rigid object can be found by using the following algorithm.

Algorithm 1. RANK MINIMIZATION BASED 3D-DEPTH RECOVERY

Data: Camera Intrinsic Parameters.
Input: (u_{ki}, v_{ki}), the temporally ordered 2-D coordinates of N points in F frames.
Output: 3D depths Z_{ki} up to an overall scaling constant.
1. Form the *difference* vectors $\mathbf{y}_k^{iN} \doteq \mathbf{P}^*_{ki} - \mathbf{P}^*_{kN},\ i = 1, \ldots, N-1$ where $\mathbf{P}^*_{ki} \doteq Z^*_{ki}\left[\frac{u_{ki}}{f}\ \frac{v_{ki}}{\alpha f}\ 1\right]^T$, and the corresponding Hankel matrices $\mathbf{H}_{\mathbf{y}^{iN}}$
2. Solve: $\min_{Z^*_{ki} \geq 1} \text{rank}\left[\mathbf{H}_{\mathbf{y}^{1N}} \ldots \mathbf{H}_{\mathbf{y}^{N-1N}}\right]$ subject to (7) and (8)

[3] This follows from simply decomposing the matrix as $\mathbf{M} = \mathbf{v}^T\mathbf{v}$, with $\mathbf{v}^T = \left[Z_{t1}\ Z_{t2}\right]$.

4.1 Computational Complexity and Robustness Considerations

In principle, Algorithm 1 will recover the unknown Z_{ij} in a single optimization step. Moreover, although rank minimization is generically NP–hard, efficient convex relaxations are available. In particular, in this paper we used the LMIRank relaxation [23]. A potential problem here is the computational cost entailed in solving simultaneously for all Z_{ki}, since the computational complexity of this relaxation scales as (number of decision variables)[5]. On the other hand, using larger sets of points minimizes the effects of outliers. To balance these effects we pursued a RANSAC (Random Sample Consensus) [24] approach. Since the minimum number of points required to define a 3D coordinate system is 4, we proceeded by finding the 3D coordinates corresponding to 4 points, randomly selected from the complete set of image points, N_s times. Out of these 4-tuples, the one preserving rigidity the most was used to find the coordinates of the remaining points by exploiting the fact that the measurements matrix has at most rank 4. Thus, given the 3D trajectories of 4 points $\mathbf{P}_{ki}$, the depth of a fifth point Z_{k5} can be found by solving a problem of the form: $\min_{\mathbf{s},Z_{k5}} \|\mathbf{W}\cdot\mathbf{s}-\mathbf{P}_5\|$, where

$$\mathbf{W}=\begin{bmatrix}\mathbf{P}_{11} & \dots & \mathbf{P}_{14}\\ \vdots & \dots & \vdots\\ \mathbf{P}_{F1} & \dots & \mathbf{P}_{F4}\end{bmatrix};\ \mathbf{P}_5 \doteq \left[\tfrac{1}{f}Z_{15}u_{15}\ \tfrac{1}{f\alpha}Z_{15}v_{15}\ Z_{15}\ \dots\ \tfrac{1}{f}Z_{F5}u_{F5}\ \tfrac{1}{f\alpha}Z_{F5}v_{F5}\ Z_{F5}\right]^T$$

5 Experiments

The accuracy of the proposed algorithm is illustrated next with experiments using synthetic and real data. In all cases, the 3D structure recovered using our algorithm (HankelSFM), is compared against the results of the Hung and Tang (HTSFM) and Mahamud and Hebert (MHSFM) algorithms. Videos of the data are provided as additional material.

5.1 Synthetic Data: The Utah Teapot

Next, we illustrate the robustness of the proposed algorithm to noise and poor calibration data. The data consists of the trajectories of the perspective projections of 137 points[4] on the Utah Teapot, centered at $(880, 250, 860)'$, as seen by a pin-hole camera with focal length $f = 400$ and image size 800×600 pixels.

In the first experiment, the teapot underwent a constant angular velocity rotation $\omega_r = 0.3$, around the axis $a = (0, 0, 1)'$, while in the second experiment, the camera is also translated with constant velocity $(-10, 5, 0)'$. Figure 1 (a)–(d) shows renderings for frames 1, 5 and 10 for the rotation experiment and the corresponding reconstructions using HankelSFM, HTSFM and MHSFM. As shown there, HankelSFM preserves the Euclidean geometry while the other methods deform the object frame to frame. Quantitative comparisons are given in Figures 1 (e)-(f), 2 and 3. Figure 1 (e)-(f) shows

[4] Nine points were selected from each surface of the Teapot.

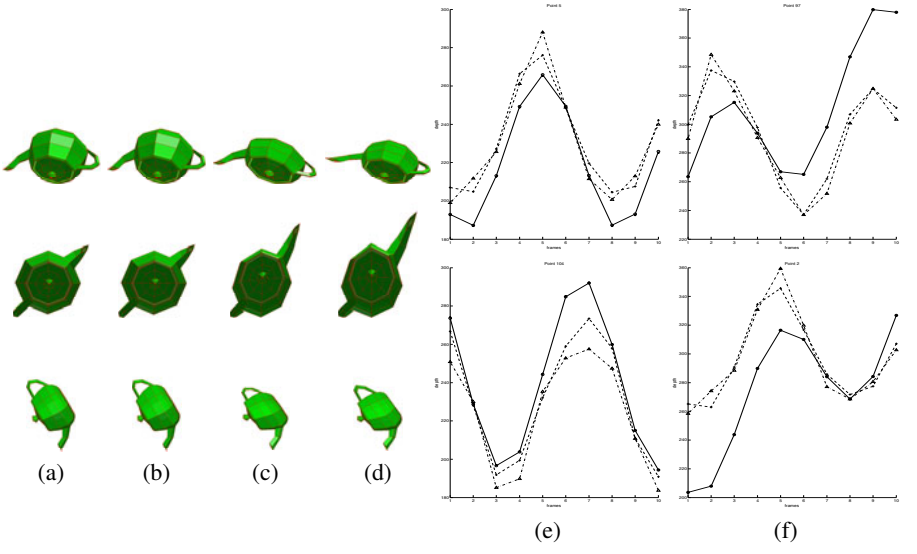
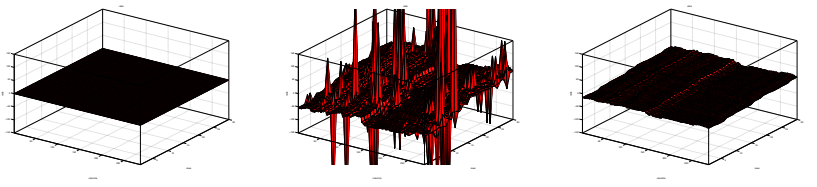


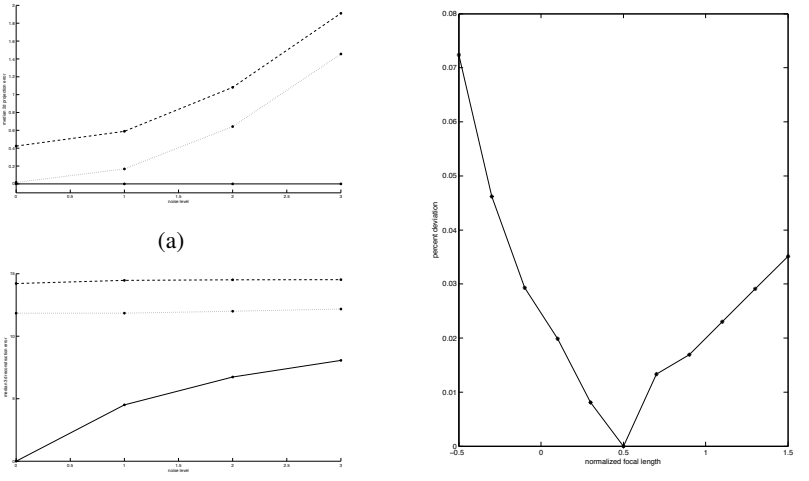
Fig. 1. (a): Frames 1, 5, and 10 of the actual teapot sequence. (b)–(d): 3D structure recovered using HankelSFM (b), MHSFM (c) and HTSFM (d). Note that HankelSFM does not introduce geometric distortion between frames. Right:Real and estimated depth trajectories for two basis points. Stars: ground truth data; solid line: HankelSFM; dashed line: MHSFM; and dashed and dotted line: HTSFM. (e) Rotation experiment. (f) Rotation and translation experiment.

the depth trajectories of two of the four points selected as basis points by the HankelSFM method, and the depths recovered using the three algorithms. All trajectories were scaled by the *single* scaling factor $c = \sum_k \sum_i Z_{ki} / \sum_k \sum_i Z^*_{ki}$ where Z_{ki} and Z^*_{ki} are the ground truth and the estimated depth for point i at frame k, respectively. Since the data is noiseless, HankelSFM exactly recovers the geometry (up to the scaling factor c) as expected, while the other methods introduce varying distortion across frames. Quantitatively, the distortion for all the points can be seen in Figure 2, showing the plots of the differences between the ratio of the elements of W and W^*, the true and reconstructed 3D measurement matrices, respectively, and the normalization factor c . As shown there, only the HankelSFM method produces a flat surface indicating a uniform scaling factor across all frames. Additionally, table 1 summarizes the 3D and the 2D re-projection median error for the three methods (noiseless data) while Figures 3 (a) and (b) plot them for increasing noise levels up to 3 pixels. In all cases, the errors are significantly lower for HankelSFM than for MHSFM and HTSFM. Finally, the very small effect of the choice of focal length on the accuracy of the depth estimation is illustrated in Figure 3 (c) where the relative variation of the scaling factor $\Delta = \max_{k,i} \|Z_{ki}/Z^*_{ki} - c\|/c$ is plotted against K, as the focal length used by the algorithm is set to Kf where f is the true focal length and $0.5 \leq K \leq 1.5$.



(a) (b) (c)

Fig. 2. $\frac{W}{W^*} - c$ for the translation and rotation Utah Teapot experiment. (a) HankelSFM. (b) MHSFM. (c) HTSFM.



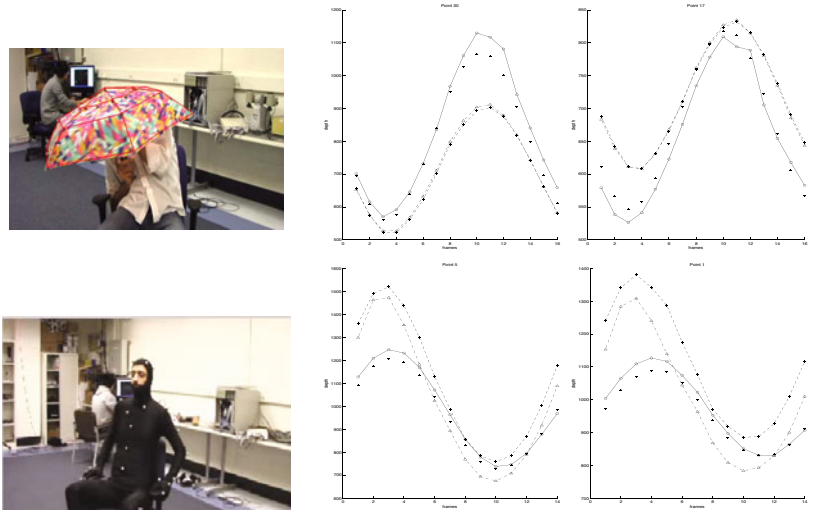
(a) (b) (c)

Fig. 3. (a) 2D re-projection and (b) 3D reconstruction median error as noise is increased from 0 to 3 pixels (solid line HankelSFM, dashed line MHSFM and dashed and dotted line HTSFM). (c) Scaling factor variation Δ as the focal length used by the algorithm is varied from 0.5 to 1.5 times the true focal length.

5.2 Real Data with Ground Truth

The purpose of these experiments is to compare the performance of HankelSFM against HTSFM and MHSFM using real data. In order to asses the accuracy of the algorithms, the 2D data was generated by projecting the *noisy* 3D coordinates of special markers attached to an umbrella and to a human sitting on a swivel chair that were measured using a VICON motion capture system[5] as shown in Figure 4, left. Quantitative results and comparisons between the 3D reconstructions and ground truth are displayed

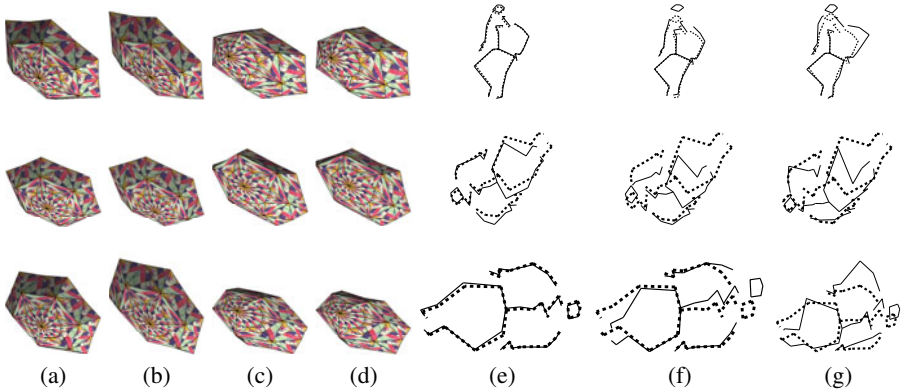
[5] It should be noted that the objects used in these experiments are flexible. Furthermore, the markers are about 1cm. in diameter and hence have a significant depth which affects the measurement of their location as the object moves in front of the motion capturing system.



(a) (b)

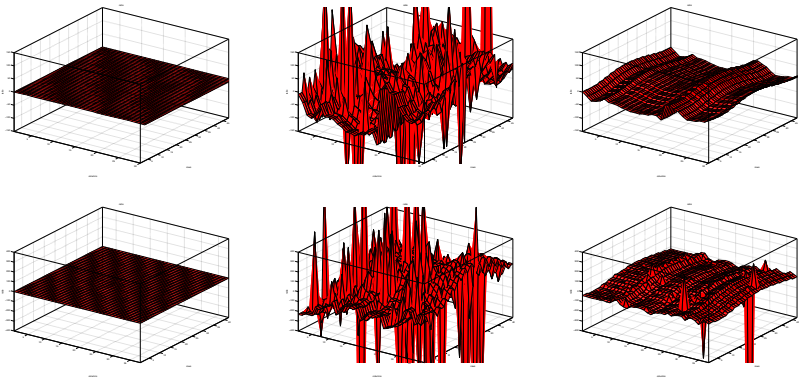
Fig. 4. (a) Sample frames of the Umbrella (top) and Human on a chair (bottom) sequences. (b) Estimated depth trajectories for two basis points. Stars: ground truth data; solid line: HankelSFM; dashed line: MHSFM; and dashed and dotted line: HTSFM.

in Figures 4, right, and 5. Finally, 3D and 2D re-projection errors are summarized in Table 1. As shown there, the HankelSFM algorithm recovers 3D structure up to a *unique* constant and its 3D accuracy outperforms the other two algorithms.



(a) (b) (c) (d) (e) (f) (g)

Fig. 5. Left: Frames 1, 6 and 12 of the umbrella sequence. (a) Ground truth data, and 3D structure recovered using (b) HankelSFM, (c) MHSFM and (d) HTSFM. Right: Frames 1, 7, 14 frames of the human on a chair sequence with ground truth data (dashed line) superimposed with 3D structure (solid line) recovered using (e) HankelSFM, (f) MHSFM and (g) HTSFM.



(a) (b) (c)

Fig. 6. $\frac{W}{W^*} - c$ for the umbrella (top row) and for the human on a chair (bottom row) sequences. (a) HankelSFM. (b) MHSFM. (c) HTSFM.

Table 1. 3D and 2D re-projection median error

Data Set	HankelSFM		MHSFM		HTSFM	
	3D (mm.)	2D (pixels^2)	3D (mm.)	2D (pixels^2)	3D (mm.)	2D (pixels^2)
Teapot (R)	4.89e-1	0	1.34e+1	3.5e+0	1.34e+1	1.2e-7
Teapot(RT)	1.61e-4	0	3.00e+1	1.0e+0	3.20e+1	2.5e-7
Umbrella	3.50e+1	0	8.22e+1	0.6176	8.32e+1	0.0136
Human	4.10e+1	0	1.37e+2	2.3091	1.51e+2	0.2713

6 Conclusions

In this paper we propose a novel algorithm for 3D Euclidean structure recovery from image sequences under perspective projection. The main idea is to exploit geometrical information encapsulated in the rank of a matrix (the Hankel matrix) constructed from the measurements. This rank implicitly encapsulates temporal information, since it strongly depends on the temporal order of the sequence: the Hankel matrices corresponding to two sequences with the same data in different order have generically different rank. The main result of the paper shows that the provably correct depths (up to an arbitrary, overall scaling constant) are the ones that minimize the rank of the corresponding Hankel matrix, thus allowing for recasting the SfM problem into a rank minimization one. This result was established by exploiting the existence of an underlying model governing the motion of the rigid body. However, no assumptions are made about this model, and there is no need to find its parameters. Indeed, our results hold independently of the object motion model. While rank-minimization problems are NP hard, recent developments in the field allow for relaxing them to a convex optimization form that can be efficiently solved. When compared to existing approaches, the

proposed algorithm recovers the 3D geometry, up to a single arbitrary scaling constant, and does require neither solving a challenging non-linear optimization, performing bundle adjustment, external camera calibration or the availability of a motion model for the moving object.

The advantages of the proposed algorithm were illustrated with synthetic and real image sequences. Research is currently underway seeking to extend these results to articulated and non-rigid objects.

References

1. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge Univeristy Press, Cambridge (2003)
2. Faugeras, O.D., Luong, Q.T., Papadopoulo, T.: The Geometry of Multiple Images. MIT Press, Cambridge (2001)
3. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. International Journal of Computer Vision 9, 137–154 (1992)
4. Morita, T., Kanade, T.: A paraperspective factorization method for recovering shape and motion from image sequences. IEEE Trans. on PAMI 19, 858–867 (1997)
5. Poelman, C.J., Kanade, T.: A paraperspective factorization method for shape and motion recovery. IEEE Transactions on PAMI 19, 206–218 (1997)
6. Sturm, P., Triggs, B.: A factorization based algorithm for multi-image projective structure and motion. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065, pp. 709–720. Springer, Heidelberg (1996)
7. Triggs, B.: Factorization methods for projective structure and motion. In: IEEE CVPR (1996)
8. Sparr, G.: Simultaneous reconstruction of scene structure and camera locations from uncalibrated image sequences. In: Int. Conf. on Pattern Recognition (1996)
9. Chen, G., Medioni, G.: Efficient iterative solutions to m-view projective reconstruction problem. In: IEEE CVPR, vol. 2, pp. 55–61 (1999)
10. Mahamud, S., Hebert, M.: Iterative projective reconstruction from multiple views. In: IEEE CVPR, vol. 2, pp. 430–437 (2000)
11. Hung, Y., Tang, W.: Projective reconstruction from multiple views with minimization of 2d reprojection error. International Journal of Computer Vision 66, 305–317 (2006)
12. Mohr, R., Veillon, F., Quan, L.: Relative 3d reconstruction using multiple uncalibrated images. In: IEEE CVPR, pp. 543–548 (1993)
13. Hartley, R.: Euclidean reconstruction from uncalibrated views. In: Mundy, J.L., Zisserman, A., Forsyth, D. (eds.) AICV 1993. LNCS, vol. 825, pp. 237–256. Springer, Heidelberg (1994)
14. Morris, D., Kanatani, K., Kanade, T.: Euclidean reconstruction from uncalibrated views. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) ICCV-WS 1999. LNCS, vol. 1883, pp. 298–375. Springer, Heidelberg (2000)
15. Shum, H.Y., Ke, Q., Zhang, Z.: Efficient bundle adjustment with virtual key frames: A hierarchical approach to multi-frame structure from motion. In: IEEE CVPR (1999)
16. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment-a modern synthesis. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) ICCV-WS 1999. LNCS, vol. 1883, pp. 298–375. Springer, Heidelberg (2000)
17. Bartoli, A., Sturm, P.: Three new algorithms for projective bundle adjustment with minimum parameters. Technical Report 4236, INRIA (2001)
18. Oliensis, J.: Fast and accurate self-calibration. In: ICCV, pp. 745–752 (1996)
19. Mahamud, S., Hebert, M., Omori, Y., Ponce, J.: Provably convergent iterative methods for projective structure frommotion. In: IEEE CVPR, pp. 1018–1025 (2001)

20. Oliensis, J., Hartley, R.: Iterative extensions of the strum/triggs algorithm: convergence and nonconvergence. IEEE Trans. on PAMI 29, 2217–2233 (2007)
21. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: Monoslam: Real time single camera slam. IEEE Trans. on PAMI 29, 1052–1067 (2007)
22. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: Proc. ISMAR 2007, Nara, Japan (November 2007)
23. Orsi, R.: LMIRank: software for rank constrained lmi problems (web page and software) (2005), http://rsise.anu.edu.au/robert/lmirank/
24. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. ACM Comm. 24, 381–395 (1981)
25. Kailath, T.: Linear Systems. Prentice-Hall, Englewood Cliffs (1980)

A Proof of Theorem 1

The proof, based on basic concepts from Linear Systems theory (see for instance the textbook [25]), consists of three steps:

1. Find an operator $\mathcal{L}$ with 2 inputs, such that its response to an impulse applied at the i^{th} input is precisely $\mathbf{y}_k^{i,\alpha_{ki}} \doteq (\alpha_{ki}\mathbf{P}_{ki} - \alpha_{k3}\mathbf{P}_{k3})$.
2. Use a realization of $\mathcal{L}$ to find the minimal rank of any linear time varying operator that interpolates the data, and to establish that the minimum rank interpolant is time-invariant and corresponds to the case $\alpha_{ki} = \lambda_o \rho^k$, for some $\lambda_o, \rho > 0$.
3. Use the connection between rank of a Linear Time Invariant (LTI) operator and the rank of its associated Hankel matrix to establish that minimizing the rank of $\mathbf{H}_{\mathbf{y}_{ki}^{\alpha}}$ recovers the depths Z_{ti} up to an overall scaling factor of the form $\alpha_t = \lambda_o \rho^t$.

Step 1; Assume[6], that the Markov parameters of $\mathcal{L}$ and $\mathbf{O}_k$ satisfy:

$$\mathbf{L}_t = \textstyle\sum_{i=1}^{n_L} \mathbf{A}_i^L \mathbf{L}_{t-i},\ \mathbf{O}_t = \sum_{i=1}^{n_O} \mathbf{A}_i^O \mathbf{O}_{t-i}, \qquad \mathbf{A}_i^L, \mathbf{A}_i^O \in R^{3\times 3} \tag{9}$$

Let $\mathbf{x}_t^i \doteq \mathbf{P}_{ti} - \mathbf{O}_t$. From the above, it follows that the trajectories $\mathbf{x}_k^i$ also satisfy a model of the form

$$\mathbf{x}_t^i = \sum_{j=1}^{n_L} \mathbf{A}_j^L \mathbf{x}_{t-j}^i, \tag{10}$$

or, in compact form:

$$\begin{aligned} \xi_{t+1}^i &= \mathcal{A}_L \xi_t^i, \\ \mathbf{x}_t^i &= \mathcal{C}_L \xi_t^i \end{aligned} \tag{11}$$

where

$$\mathcal{A}_L \doteq \begin{bmatrix} \mathbf{A}_1^L & \mathbf{A}_2^L & \dots & \mathbf{A}_{n_L-1}^L & \mathbf{A}_{n_L}^L \\ \mathbf{I} & 0 & \dots & \dots & 0 \\ 0 & \mathbf{I} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{I} & 0 \end{bmatrix} \quad \xi_t^i \doteq \begin{bmatrix} \mathbf{x}_{t-1}^i \\ \mathbf{x}_{t-2}^i \\ \vdots \\ \mathbf{x}_{t-n_L}^i \end{bmatrix}, \ \mathcal{C}_L = \begin{bmatrix} \mathbf{I}\, \mathbf{0} \dots \mathbf{0} \end{bmatrix}$$

[6] This is without loss of generality, since over finite horizons, any trajectory $\mathbf{L}_k$ can be interpolated with an ARMA model of sufficiently high order.

With this notation, the trajectories $\mathbf{x}_t^i$ in (10) are given by:

$$\mathbf{x}_t^i = \mathcal{C}_L \xi_t^i = \mathcal{C}_L \mathcal{A}_L \xi_{t-1}^i = \cdots = \mathcal{C}_L \mathcal{A}_L^t \xi_o^i \tag{12}$$

Thus, $\mathbf{x}_t^i$ is the impulse response of the system:

$$\begin{aligned} \xi_{t+1}^i &= \mathcal{A}_L \xi_t^i + \xi_o^i \delta_t \\ \mathbf{x}_t^i &= \mathcal{C}_L \xi_t^i \end{aligned} \tag{13}$$

A similar situation holds for $\mathbf{O}_t$, with $\mathbf{A}_L^j$ and $\mathcal{A}_L$ replaced by $\mathbf{A}_O^j$ and $\mathcal{A}_O$, respectively, and ξ_t by a vector ω_t containing the past values $\mathbf{O}_k,\ k = t, \ldots, t - n_O + 1$. Hence $\mathbf{O}_t$ can be obtained as the impulse response of a system with state space realization $(\mathcal{A}_O, \omega_o, [\mathbf{I}\,\mathbf{0}\ \ldots\ \mathbf{0}])$.

Given two points $\mathbf{P}_i, \mathbf{P}_j$ from the rigid, and a time varying scaling constant α_t, consider now the vector $\mathbf{y}_t^{\alpha_t} \doteq (\alpha_t \mathbf{P}_{ti} - \mathbf{P}_{tj})$. Since $\mathbf{P}_{ti} = \mathbf{x}_t^i + \mathbf{O}_t$, we have that

$$\mathbf{y}_t^{\alpha_t} = \alpha_t(\mathbf{x}_t^i + \mathbf{O}_t) - (\mathbf{x}_t^j + \mathbf{O}_t) = \alpha_t \mathbf{x}_t^i - \mathbf{x}_t^j + (\alpha_t - 1)\mathbf{O}_t$$

From (13) and linearity it follows that the trajectory $\mathbf{y}_t^{\alpha_t}$ can be generated as the impulse response of the system:

$$\zeta_{t+1} = \begin{bmatrix} \mathcal{A}_L & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathcal{A}_L & \mathbf{0} \\ \mathbf{O} & \mathbf{O} & \mathcal{A}_O \end{bmatrix} \zeta_t + \begin{bmatrix} \xi_o^i \\ \xi_o^j \\ \omega_o \end{bmatrix} \delta_t \tag{14}$$

$$\mathbf{y}_t^{\alpha_t} = \begin{bmatrix} \alpha_t \mathcal{C}_L & -\mathcal{C}_L & (\alpha_t - 1)\mathcal{C}_O \end{bmatrix} \zeta_t$$

Finally, consider three points $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$ and the corresponding vectors $\mathbf{y}^{i\alpha_{ti}} \doteq \alpha_{ti}\mathbf{P}_{ti} - \alpha_{t3}\mathbf{P}_{t3},\ i = 1, 2$. It follows from above that the two trajectories $\mathbf{y}^{i\alpha_{ti}}$ can be simultaneously generated as the impulse response of the system:

$$\begin{aligned} \zeta_{t+1} &= \mathcal{A}\zeta_t + \mathcal{B}u; \quad u \in R^2 \\ y_t &= \mathcal{C}_t \zeta_t \end{aligned} \tag{15}$$

where

$$\mathcal{A} = \begin{bmatrix} \mathcal{A}_L & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathcal{A}_L & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathcal{A}_O & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathcal{A}_L & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathcal{A}_L & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathcal{A}_O \end{bmatrix} \quad \mathcal{B} = \begin{bmatrix} \xi_o^1 & 0 \\ \xi_o^3 & 0 \\ \omega_o & 0 \\ 0 & \xi_o^2 \\ 0 & \xi_o^3 \\ 0 & \omega_o \end{bmatrix}, \quad \mathcal{C}_L = \begin{bmatrix} \mathbf{I}\,0 \ldots 0 \end{bmatrix}, \quad \mathcal{C}_O = \begin{bmatrix} \mathbf{I}\,0 \ldots 0 \end{bmatrix}$$

$$\mathcal{C}_t = [\alpha_{t1}\mathcal{C}_L\ \alpha_{t3}\mathcal{C}_L\ (\alpha_{t1} - \alpha_{t3})\mathcal{C}_O\ \alpha_{t2}\mathcal{C}_L\ -\alpha_{t3}\mathcal{C}_L\ (\alpha_{t2} - \alpha_{t3})\mathcal{C}_O] \tag{16}$$

Step 2: Recall [25] that for linear time invariant systems, given a triple $(\mathcal{A}, \mathcal{B}, \mathcal{C})$, with $\mathcal{A} \in R^{n \times n}$, the order of the minimal realization $(\mathcal{A}_m, \mathcal{B}_m, \mathcal{C}_m)$ that has the same input/output response is given by the rank of the product of its controllability and observability matrices, defined as:

$$\mathcal{K}_{ctrb} = \begin{bmatrix} \mathcal{B}\ \mathcal{A}\mathcal{B} \ldots \mathcal{A}^{n-1}\mathcal{B} \end{bmatrix}, \quad \mathcal{K}_{obs} = \begin{bmatrix} \mathcal{C}^T\ \mathcal{A}^T\mathcal{C}^T \ldots (\mathcal{A}^{n-1})^T\mathcal{C}^T \end{bmatrix} \tag{17}$$

However, this result cannot be directly applied to (15), due to the time–varying scaling factors α_{ti} in $\mathcal{C}_t$. In this case, the order of the minimal realization $(\mathcal{A}_m, \mathcal{B}_m, \mathcal{C}_m)$ that has the same input/output response as the original triple $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ is given by ([25], Chapter 9) $\text{rank}(W_t^c W_t^o)$ where

$$\mathbf{W}_t^o = (\mathcal{K}_{t,o})^T \mathcal{K}_{t,o},\ \mathbf{W}_t^c = (\mathcal{K}_{t,c})^T \mathcal{K}_{t,c},\ \mathcal{K}_{t,o} = \begin{bmatrix} \mathcal{C}_{t-1} \\ \mathcal{C}_{t-2}\mathcal{A} \\ \vdots \\ \mathcal{C}_o\mathcal{A}^{t-1} \end{bmatrix},\ \mathcal{K}_{t,c} = \begin{bmatrix} \mathcal{B}\ \mathcal{A}\mathcal{B} \dots \mathcal{A}^{t-1}\mathcal{B} \end{bmatrix}$$

Note that the pair $(\mathcal{A}, \mathcal{B})$ is time invariant (since no scaling factors are involved). Further, from a PBH argument (see [25], page 366) it can be shown that, if $t \geq n$, then, generically, $\text{rank}(\mathcal{K}_{t,c}) = n$. On the other hand, using the explicit expressions for $\mathcal{A}$ and $\mathcal{C}$ yields, for each block-row of $\mathcal{K}_{t,o}$:

$$(\mathcal{K}_{t,o})_j = \begin{bmatrix} \alpha_{(t-j)1} \left(K_{obs}^L\right)_j & -\alpha_{(t-j)3} \left(K_{obs}^L\right)_j & (\alpha_{(t-j)1} - \alpha_{(t-j)3}) \left(K_{obs}^O\right)_j \\ \alpha_{(t-j)2} \left(K_{obs}^L\right)_j & -\alpha_{(t-j)3} \left(K_{obs}^L\right)_j & (\alpha_{(t-j)2} - \alpha_{(t-j)3}) \left(K_{obs}^O\right)_j \end{bmatrix}$$

where $(\mathbf{M})_j$ denotes the j^{th} block–row of a matrix $\mathbf{M}$, and K_{obs}^L, K_{obs}^O denote the observability matrices of $(\mathcal{C}_L, \mathcal{A}_L)$ and $(\mathcal{C}_O, \mathcal{A}_O)$, respectively. Since by construction both realizations are observable, it follows that, if the motion of O_k has at least one mode not contained in the operator $\mathcal{L}$ (the relative motion of the rigid with respect to O) then the minimum rank of $\mathcal{K}_{t,o}$ over all $\alpha_{ti} > 0$ is achieved by selecting $\alpha_{t1} = \alpha_{t2} = \alpha_{t3} = \alpha_t$, an overall, time varying scaling factor. Further, note that this minimum is achieved by an LTI system if and only if $\alpha_t = \lambda_o \rho^t$ for some $\lambda_o, \rho \neq 0$.

Step 3. Let $\hat{Z}_{ti}$ and $\hat{\mathbf{P}}_{ti}$, denote the actual values of Z_{ti} and the 3D trajectories, respectively. Consider any candidate trajectory $\tilde{Z}_{ti} \doteq \alpha_{ti} \hat{Z}_{ti}$ and denote by $\mathbf{P}_{ti}$, the 3D trajectory reconstructed from the 2D data using $\tilde{Z}_{ti}$. Finally, define the difference vectors:

$$\mathbf{y}_t^i \doteq \mathbf{P}_{ti} - \mathbf{P}_{t3} = \left(\alpha_{ti} \hat{\mathbf{P}}_{ti} - \alpha_{t3} \hat{\mathbf{P}}_{t3}\right) \tag{18}$$

and the associated matrix $\mathbf{H}_y = \left[\mathbf{H}_{\mathbf{y}^1}\ \mathbf{H}_{\mathbf{y}^2}\right]$. Consider any sequence $\tilde{\alpha}_{ti} > 0$ and let $\mathcal{L}(\tilde{\alpha}_{ti})$ denote the associated operator. From step 2 above, it follows that

$$min_{\alpha_{ti}} \text{rank}\{\mathcal{L}(\alpha_{ti})\} \leq \text{rank}\{\mathcal{L}(\tilde{\alpha}_{ti})\} \leq \text{rank}\{\mathbf{H}(\tilde{\alpha}_{ti})\}$$

with the equalities holding only in the case where $\mathcal{L}$ is an LTI operator, e.g. $\tilde{\alpha}_{ti} = \lambda_o \rho^t,\ i = 1, 2, 3$. Hence, the depths Z_{ti} obtained by minimizing the rank of $\mathbf{H}_y$ satisfy $Z_{ti} = \lambda_o \rho^t \hat{Z}_{ti}$ for some $\lambda_o, \rho \neq 0$.

Exploiting Loops in the Graph of Trifocal Tensors for Calibrating a Network of Cameras

Jérôme Courchay[1], Arnak Dalalyan[1], Renaud Keriven[1], and Peter Sturm[2]

[1] IMAGINE, LIGM, Université Paris-Est
[2] Laboratoire Jean Kuntzmann, INRIA Grenoble Rhône-Alpes

Abstract. A technique for calibrating a network of perspective cameras based on their graph of trifocal tensors is presented. After estimating a set of reliable epipolar geometries, a parameterization of the graph of trifocal tensors is proposed in which each trifocal tensor is encoded by a 4-vector. The strength of this parameterization is that the homographies relating two adjacent trifocal tensors, as well as the projection matrices depend linearly on the parameters. A method for estimating these parameters in a global way benefiting from loops in the graph is developed. Experiments carried out on several real datasets demonstrate the efficiency of the proposed approach in distributing errors over the whole set of cameras.

1 Introduction

Camera calibration from images of a 3-dimensional scene has always been a central issue in Computer Vision. The success of textbooks like [1,2] attests this interest. In recent years, many methods for calibration have been proposed. Most of these work either rely on known or partially known internal calibrations [3,4,5,6,7,8,9,10] or deal with an ordered sequence of cameras [11,12,13,14]. In many practical situations, however, the internal parameters of cameras are unavailable or available but very inaccurate. The absence of an order in the set of cameras is also very common when processing, for instance, Internet images.

In this paper, we deal with the problem of calibrating a network of cameras from a set of unordered images, the main emphasis being on the accuracy of the projective reconstruction of camera matrices. Traditionally, this situation is handled by factorizing the measurement matrix [15,16], which may be subject to missing data [17,18] because of occlusions. The methodology adopted in the present work is substantially different and is based on the notion of the graph of trifocal tensors rather than on the factorization. The experiments on real datasets show that our approach leads to highly competitive results that furnish a good initialization to the bundle adjustment (BA) algorithm [19].

Even in the case of calibrated cameras, most of the aforementioned methods are based on a graph of cameras (in which the edges are the epipolar geometries) which is made acyclic by discarding several edges. On the other hand, a number of recent studies, oriented toward city modeling from car or aerial sequences, point out the benefits of enforcing loop constraints. Considering loops in the graph of cameras has the advantage of reducing the drift due to errors induced while processing the trajectory sequentially

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 85–99, 2010.

(cf. Fig. 1). [20] merges partial reconstructions, [21] constrains coherent rotations for loops and planar motion. Adapted to their specific input, these papers often rely on trajectory regularization or dense matching [22,23]. [24] is a notable exception, where loop constraints are added to sparse Structure from Motion (SfM), yet taking as input an ordered omnidirectional sequence and assuming known internal parameters. The

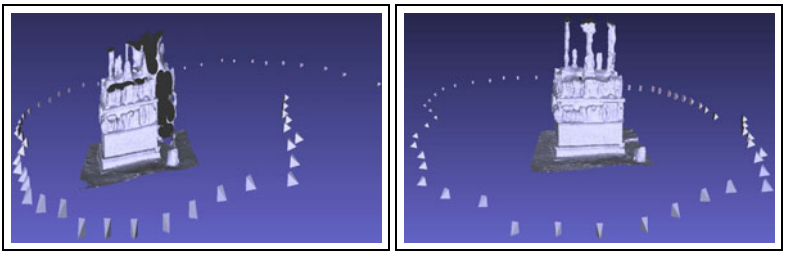


Fig. 1. Multi-view stereo reconstruction [25] using cameras calibrated without (left) and with (right) using the loop-constraint. When the loop constraint is not enforced, the accumulation of errors results in an extremely poor reconstruction.

method proposed in the present work consists of the following points:

- Our starting point is a set of unknown cameras linked by estimated epipolar geometries (EG). These are computed using a state-of-the-art version of RANSAC [26], followed by a maximum likelihood improvement described in [13]. We assume that along with the estimated fundamental matrices, reliable epipolar correspondences are known. These correspondences are made robust by simultaneously considering several camera pairs, like in [3]. This produces a set of three-view correspondences that will be used in the sequel.
- We group views into triplets. Three views (i, j, k) are considered as a valid triplet if (a) the EGs between i and j as well as between j and k have been successfully computed at the previous step and (b) there are at least 4 three-view correspondences in these images. To reduce the number of nodes, some of the estimated epipolar geometries are ignored, so that inside a triplet, only two of the three fundamental matrices are considered known. The advantage of this strategy is that we do not need to enforce the coherence of fundamental matrices. At first sight, this can be seen as a loss of information. However, this information is actually recovered via trifocal tensors.
- We define a graph having as nodes valid camera triplets. Therefore, there are two fundamental matrices available for each node. Two nodes are connected by an edge if they share a fundamental matrix. We demonstrate that for each node there exists a 4-vector such that all the entries of the three camera matrices are affine functions of this 4-vector with known coefficients. Moreover, the homographies that allow the registration of two adjacent nodes ν and ν' are also affine functions depending on 4 out of the 8 unknown parameters corresponding to ν and ν'. To speed-up the computations, for each node only 50 (or less) three-view correspondences that are the most compatible with the EGs are used.

– If the graph of triplets is acyclic, the equations of three-view correspondences for all nodes lead to a linear estimate of all the cameras. In case, the graph of triplets contains one or several loops, each loop is encoded as a (non-linear) constraint on the unknowns. Starting from an initial value computed as a solution to the unconstrained least squares, we sequentially linearize the loop constraints and solve the resulting problem by (sparse) linear programming. This can be efficiently done even for very large graphs. It converges very rapidly, but the loop constraints are fulfilled only approximately.
– In the case where the loop constraints are not satisfied exactly, we proceed by homography registration and estimation of camera matrices by linear least-squares under norm constraint. This is done exactly via a singular value decomposition producing as output all cameras in a projective space. To provide a qualitative evaluation, we recover the metric space using an implementation of [27], and a single Euclidean bundle adjustment that refines the metric space and camera positions.

Thus, we propose a method that accurately recovers geometries, without any sequential process, and attempts to enforce the compatibility of cameras within loops in the early stages of the procedure. An important advantage conferred by our approach is that the number of unknown parameters is kept fairly small, since we consider only the cameras (four unknowns for each triplet) and not the 3D points. Our reconstruction is further refined by bundle adjustment. Taking loops into account and avoiding error accumulation, the proposed solution is less prone to get stuck in local minima.

The remainder of the paper is organized as follows. Section 2 presents the background theory and terminology. Our algorithm is thoroughly described in Sections 3 and 4. The results of numerical experiments conducted on several real datasets as well as a comparison to state-of-the-art software is provided in Section 5. A discussion concludes the paper.

2 Background

In this work, we consider a network of N uncalibrated cameras and assume that for some pairs of cameras (i, j), where $i, j = 1, \ldots, N$, $i \neq j$, an estimation of the fundamental matrix, denoted by $\mathtt{F}^{ij}$, is available. Let us denote by $\mathbf{e}^{ij}$ the unit norm epipole in view j of camera center i. Recall that the fundamental matrix leads to a projective reconstruction of camera matrices $(\mathtt{P}^i, \mathtt{P}^j)$, which is unique up to a homography.

The geometry of three views i, j and k is described by the Trifocal Tensor, hereafter denoted by $\mathcal{T}^{ijk}$. It consists of three 3×3 matrices: $\mathtt{T}_1^{ijk}$,$\mathtt{T}_2^{ijk}$ and $\mathtt{T}_3^{ijk}$ and provides a particularly elegant description of point-line-line correspondences in terms of linear equations

$$\mathbf{p}_i^\mathsf{T} \begin{bmatrix} \mathbf{l}_j^\mathsf{T} \mathtt{T}_1^{ijk} \\ \mathbf{l}_j^\mathsf{T} \mathtt{T}_2^{ijk} \\ \mathbf{l}_j^\mathsf{T} \mathtt{T}_3^{ijk} \end{bmatrix} \mathbf{l}_k = 0, \tag{1}$$

where $\mathbf{p}_i$ is a point in image i (seen as a point in projective space $\mathbb{P}^2$) which is in correspondence with the line $\mathbf{l}_j$ in image j and with the line $\mathbf{l}_k$ in image k. Considering

the entries of $\mathcal{T}^{ijk}$ as unknowns, we get thus one linear equation for each point-line-line correspondence. Therefore, one point-point-point correspondence $\mathbf{p}_i \longleftrightarrow \mathbf{p}_j \longleftrightarrow \mathbf{p}_k$ leads to 4 independent linear equations by combining an independent pair of lines passing through $\mathbf{p}_j$ in image j with an independent pair of lines passing through $\mathbf{p}_k$ in image k.

Since a Trifocal Tensor has 27 entries, the previous argument shows that 7 point-point-point correspondences suffice for recovering the Trifocal Tensor as a solution of an overdetermined system of linear equations. Recall however that the Trifocal Tensor has only 18 degrees of freedom. Most algorithms estimating a Trifocal Tensor from noisy point-point-point correspondences compute an approximate solution to the linear system by a least squares estimator (LSE) and then perform a post-processing in order to get a valid Trifocal Tensor. An alternative approach consists in using a minimal solution that determines the three-view geometry from six points [28,29].

2.1 Main Ingredients of Our Approach

Let us describe now two elementary results that represent the building blocks of our approach, relying on the fact that when two out of three fundamental matrices are known, the Trifocal Tensor has exactly 4 degrees of freedom.

Proposition 1. *For three views i, j and k, given two fundamental matrices F^{ij} and F^{ik}, there exists a 4-vector $\boldsymbol{\gamma} = [\gamma_0, \ldots, \gamma_3]$ such that $\mathcal{T}^{ijk}$ is given by:*

$$T_t^{ijk} = A_t^{ij} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & \gamma_0 & \gamma_t \end{bmatrix} \left(A_t^{ik}\right)^{\mathsf{T}} \tag{2}$$

for every $t = 1, 2, 3$, where $A_t^{is} = \left[(F_{t,1:3}^{is})^{\mathsf{T}} \ , \ (F_{t,1:3}^{is})^{\mathsf{T}} \times \mathbf{e}^{is} \ , \ \mathbf{e}^{is}\right]$, for $s = j, k$. Moreover, $\mathcal{T}^{ijk}$ is geometrically valid, i.e. , there exist 3 camera matrices P^i, P^j and P^k compatible with F^{ij} and F^{ik} and having $\mathcal{T}^{ijk}$ as the Trifocal Tensor.

The proof of this result is deferred to the supplemental material. It is noteworthy that the claims of Proposition 1 hold true under full generality, even if the centers of three cameras are collinear. In view of [1], the camera matrices parameterized by $\boldsymbol{\gamma}$ that are compatible with the fundamental matrices F^{ij} and F^{ik} as well as with the Trifocal Tensor defined by Eq. 2 are given by (up to a projective homography)

$$\begin{aligned} \mathrm{P}^i &= [\mathrm{I}_{3\times 3} \,|\, \mathbf{0}_{3\times 1}], \qquad \mathrm{P}^k = [\gamma_0 [\mathbf{e}^{ik}]_\times \mathrm{F}^{ki} \,|\, \mathbf{e}^{ik}], \\ \mathrm{P}^j &= \mathrm{kron}\big([\boldsymbol{\gamma}_{1:3}, 1]; \mathbf{e}^{ij}\big) - [[\mathbf{e}^{ij}]_\times \mathrm{F}^{ji} \,|\, \mathbf{0}_{3\times 1}], \end{aligned} \tag{3}$$

where $\mathrm{kron}(\cdot, \cdot)$ stands for the Kronecker product of two matrices.

In the noiseless setting, Proposition 1 offers a minimal way of computing the 4 remaining unknowns from point-point-point correspondences. One could think that one point-point-point correspondence leading to 4 equations is enough for retrieving the 4 unknowns. However, since two EGs are known, only one equation brings new information from one point-point-point correspondence. So we need at least 4 point-point-point

correspondences to compute the Trifocal Tensor compatible with the two given fundamental matrices. In the noisy case, if we use all 4 equations associated to point-point-point correspondences, the system is then overdetermined and one usually proceeds by computing the LSE.

The second ingredient in our approach is the parameterization of the homography that bridges two camera triplets having one fundamental matrix in common. Let i, j, k and ℓ be four views such that (a) for views i and k we have successfully estimated the fundamental matrix $\mathtt{F}^{ik}$ and (b) for each triplet (i,j,k) and (k,i,ℓ) the estimates of two fundamental matrices are available. Thus, the triplets (i,j,k) and (k,i,ℓ) share the same fundamental matrix $\mathtt{F}^{ik}$. Using equations (3), one obtains two projective reconstructions of camera matrices of views i and j based on two 4-vectors $\boldsymbol{\gamma}$ and $\boldsymbol{\gamma}'$. Let us denote the reconstruction from the triplet (i,j,k) (resp. (k,i,ℓ)) by $\mathtt{P}^i_{\boldsymbol{\gamma}}$ and $\mathtt{P}^k_{\boldsymbol{\gamma}}$ (resp. $\mathtt{P}^i_{\boldsymbol{\gamma}'}$ and $\mathtt{P}^k_{\boldsymbol{\gamma}'}$). If the centers of cameras i and k differ, then there is a unique homography $\mathtt{H}_{\boldsymbol{\gamma},\boldsymbol{\gamma}'}$ such that

$$\mathtt{P}^i_{\boldsymbol{\gamma}}\mathtt{H}_{\boldsymbol{\gamma},\boldsymbol{\gamma}'} \cong \mathtt{P}^i_{\boldsymbol{\gamma}'}, \quad \mathtt{P}^k_{\boldsymbol{\gamma}}\mathtt{H}_{\boldsymbol{\gamma},\boldsymbol{\gamma}'} \cong \mathtt{P}^k_{\boldsymbol{\gamma}'}, \tag{4}$$

where $\cong$ denotes equality up to a scale factor. Considering the camera matrices as known, one can solve (4) w.r.t. $\mathtt{H}_{\boldsymbol{\gamma},\boldsymbol{\gamma}'}$. One readily checks that[1]

$$\mathtt{H}_{\boldsymbol{\gamma},\boldsymbol{\gamma}'} = \left[\begin{array}{c|c} \mathrm{kron}(\boldsymbol{\gamma}'_{1:3}, \mathbf{e}^{ki}) - [\mathbf{e}^{ki}]_\times \mathtt{F}^{ik} & \mathbf{e}^{ki} \\ \hline -\frac{\gamma_0}{2}\,\mathrm{tr}([\mathbf{e}^{ik}]_\times \mathtt{F}^{ki}[\mathbf{e}^{ki}]_\times \mathtt{F}^{ik})(\mathbf{e}^{ik})^\mathsf{T} & 0 \end{array}\right]. \tag{5}$$

To sum up this section, let us stress that the main message to retain from all these formulas is that $\mathtt{H}_{\boldsymbol{\gamma},\boldsymbol{\gamma}'}$, as well as the camera matrices (3) are linear in $(\boldsymbol{\gamma},\boldsymbol{\gamma}')$.

3 Estimating Tensors by Sequential Linear Programming

This section contains the core of our contribution which is based on a graph-based representation of the triplets of cameras. This is closely related to the framework developed in [5], where the graph of camera pairs is considered. The advantage of operating with triplets instead of pairs is that there is no need to distinguish between feasible and infeasible paths.

3.1 Graph of Trifocal Tensors

The starting point for our algorithm is a set of estimated EGs that allow us to define a graph $\mathcal{G}_{\text{cam}}$ so that (a) $\mathcal{G}_{\text{cam}}$ has N nodes corresponding to the N cameras and (b) two nodes of $\mathcal{G}_{\text{cam}}$ are connected by an edge if a reliable estimation of the corresponding epipolar geometry is available. Then, a triplet of nodes i,j,k of $\mathcal{G}_{\text{cam}}$ is called valid if (a) there is a sufficient number of three view correspondences between i, j and k, and (b) at least two out of three pairs of nodes are adjacent in $\mathcal{G}_{\text{cam}}$.

If for some valid triplet all three EGs are available, we remove the least reliable one and define the graph $\mathcal{G}_{\text{triplet}} = (\mathcal{V}_{\text{triplet}}, \mathcal{E}_{\text{triplet}})$ having as nodes valid triplets of cameras

[1] See supplemental material for more details.

and as edges the pairs of triplets that have one fundamental matrix in common. In view of Proposition 1, the global calibration of the network is equivalent to the estimation of a 4-vector for each triplet of cameras. Thus, to each node v of the graph of triplets we associate a vector $\boldsymbol{\gamma}^v \in \mathbb{R}^4$. The large vector $\Gamma = (\boldsymbol{\gamma}^v : v \in \mathcal{V}_{\text{triplet}})$ is the parameter of interest in our framework.

If, by some chance, it turns out that the graph of triplets is acyclic, then the problem of estimating Γ reduces to estimating $N_V = \text{Card}(\mathcal{V}_{\text{triplet}})$ independent vectors $\boldsymbol{\gamma}^v$. This task can be effectively accomplished using point-point-point correspondences and the equation (1). As explained in Section 2, a few point-point-point correspondences suffice for computing an estimator of $\boldsymbol{\gamma}^v$ by least squares. In our implementation, we use RANSAC with a minimal configuration of four 3-view correspondences in order to perform robust estimation.

3.2 Calibration as Constrained Optimization

However, acyclic graphs are the exception rather than the rule. Even if the camera graph is acyclic, the resulting triplet graph may contain loops. To explain how the loops in the graph $\mathcal{G}_{\text{triplet}}$ are handled, let us remark that one can associate a homography (cf. (5)) to each adjacent pair (v, v') of nodes of $\mathcal{G}_{\text{triplet}}$. Using these homographies, each loop of the graph of triplets yields a constraint on the homographies and, therefore, on the parameter vector Γ. For instance, the 3-loop $v \longleftrightarrow v' \longleftrightarrow v'' \longleftrightarrow v$ gives rise to the constraint $\mathtt{H}_{\boldsymbol{\gamma}^v,\boldsymbol{\gamma}^{v'}}\mathtt{H}_{\boldsymbol{\gamma}^{v'},\boldsymbol{\gamma}^{v''}}\mathtt{H}_{\boldsymbol{\gamma}^{v''},\boldsymbol{\gamma}^{v}} \cong \mathtt{I}_{4\times 4}$. This equation defines a set of 15 polynomial constraints on the unknown vector Γ. If the triplet graph contains N_{loop} loops, then we end up with $15N_{\text{loop}}$ constraints. Our proposal—in the case of general graphs of triplets—is to estimate Γ by minimizing an energy derived from the equations (1) and point-point-point correspondences (similarly to the LSE proposed in the previous subsection) subject to $15N_{\text{loop}}$ constraints.

The main advantage of this approach is that if a solution to the proposed optimization problem is found, it is guaranteed to be consistent w.r.t. the loops, meaning that each camera matrix will be uniquely determined up to a scale factor and an overall homography ambiguity.

3.3 Sequential Linear Programming

Instead of solving the optimization problem that is obtained by combining the LSE with the loop-constraints, we propose here to replace it by a linear program. To give more details, let us remark that every loop-constraint can be rewritten as $f_j(\Gamma) = 0$, $j = 1, \ldots, 15$, for some polynomial functions f_j. Gathering these constraints for all N_{loop} loops, we get

$$f_j(\Gamma) = 0, \quad j = 1, \ldots, 15N_{\text{loop}}. \tag{6}$$

On the other hand, in view of (1) and (2), the point-point-point correspondences can be expressed as an inhomogeneous linear equation system in Γ

$$\mathtt{M}\Gamma = \mathbf{m}, \tag{7}$$

where $\mathtt{M}$ is a $4N_{\text{3-corr}} \times 4N$ matrix and $\mathbf{m}$ is a $4N_{\text{3-corr}}$ vector with $N_{\text{3-corr}}$ being the number of correspondences across three views. The matrix $\mathtt{M}$ and the vector $\mathbf{m}$ are computed using the known fundamental matrices. Since in practice these matrices are estimated from available data, the system (7) need not be satisfied exactly. Then, it is natural to estimate the parameter-vector Γ by solving the problem

$$\min \|\mathtt{M}\Gamma - \mathbf{m}\|_q^q \qquad \text{subject to} \quad f_j(\Gamma) = 0,\ \forall j = 1, \ldots, 15N_{\text{loop}}, \tag{8}$$

for some $q \geq 1$. Unfortunately, there is no q for which this problem is convex and, therefore, it is very hard to solve. To cope with this issue, we propose a strategy based on local linearization.

We start by computing an initial estimator of Γ, *e.g.*, by solving the unconstrained (convex) problem with some $q \geq 1$. In our implementation, we use RANSAC with $q = 2$ for ensuring robustness to erroneous three-view correspondences. Then, given an initial estimator Γ_0, we define the sequence Γ_k by the following recursive relation: Γ_{k+1} is the solution to the linear program

$$\min \|\mathtt{M}\Gamma - \mathbf{m}\|_1 \qquad \text{subject to} \quad |f_j(\Gamma_k) + \nabla f_j(\Gamma_k)(\Gamma - \Gamma_k)| \leq \epsilon, \tag{9}$$

where ϵ is a small parameter (we use $\epsilon = 10^{-6}$). There are many softwares—such as GLPK, SeDuMi, SDP3—for solving problem (9) with highly attractive execution times even for thousands of constraints and variables. Furthermore, empirical experience shows that the sequence Γ_k converges very rapidly. Typically, a solution with satisfactory accuracy is obtained after five to ten iterations.

3.4 Accounting for Heteroscedasticity

The goal now is to make the energy that we minimize in (9), which is purely algebraic, meaningful from a statistical viewpoint. Assume equations (7) are satisfied up to an additive random noise: $\mathtt{M}\Gamma = \mathbf{m} + \boldsymbol{\xi}$, where the random vector $\boldsymbol{\xi}$ has independent coordinates drawn from the centered Laplace distribution with constant scale. Then the energy in (9) is proportional to the negative log-likelihood. The constancy of the scale factor means that the errors are homoscedastic, which is a very strong hypothesis. We observed that all three view correspondences recorded by a fixed triplet have nearly the same scale for the errors, while the scales for different triplets are highly variable. To account for this heteroscedasticity of the noise, we use the initial estimator of Γ to estimate one scale parameter σ_v per node $v \in \mathcal{V}_{\text{triplet}}$. This is done by computing the standard deviation of the estimated residuals. Using $\{\sigma_v\}$, the energy in problem (9) is replaced by $\sum_v \|\mathtt{M}_v\Gamma - \mathbf{m}_v\|_1/\sigma_v$. Here, $\mathtt{M}_v$ is the submatrix of $\mathtt{M}$ containing only those rows that are obtained from three-view correspondences recorded by v. The vector $\mathbf{m}_v$ is obtained from $\mathbf{m}$ in the same way.

4 Homography Registration and Estimation of Projection Matrices

Assume that we have a graph of trifocal tensors, $\mathcal{G}_{\text{triplet}}$, each node of which will be denoted by $v_1, v_2, \ldots, v_n$. In the previous step, we have determined parameters $\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_n$,

such that $\boldsymbol{\gamma}_i$ characterizes the trifocal tensor represented by v_i. A naive strategy for estimating camera matrices is to set one of the cameras equal to $[\mathtt{I}_{3\times 3} \,|\, \mathbf{0}_{3\times 1}]$ and to recover the other cameras by successive applications of the homographies $\mathtt{H}_{\gamma,\gamma'}$ to the camera matrices reconstructed according to (3). However, in general situations, the vector Γ computed by sequential linear programming as described in the previous section satisfies the loop constraints up to a small error. Therefore, the aforementioned naive strategy has the drawback of increasing the error of estimation for cameras computed using many homographies $\mathtt{H}_{\gamma,\gamma'}$. In order to avoid this and to uniformly distribute the estimation error over the set of camera matrices, we propose a method based on homography registration by SVD. Thus, the input for the method described in this section is a vector Γ for which the loop constraints are satisfied up to a small estimation error.

4.1 The Case of a Single Loop

We assume in this subsection that $\mathcal{G}_{\text{triplet}}$ reduces to one loop, that is each node v_i has exactly two neighbors v_{i-1} and v_{i+1} with standard convention and $v_{n+i} = v_i$ for all i. (This applies to all the indices in this subsection.) For each node v_i representing three views, we have already computed a version of the projection matrices $\mathtt{P}^{1,\boldsymbol{\gamma}_i}, \mathtt{P}^{2,\boldsymbol{\gamma}_i}, \mathtt{P}^{3,\boldsymbol{\gamma}_i}$. Furthermore, for two neighboring nodes v_i and v_{i+1} we have computed a homography $\mathtt{H}^{i,i+1}$ so that $\mathtt{P}^{j,\boldsymbol{\gamma}_{i+1}} \cong \mathtt{P}^{j+1,\boldsymbol{\gamma}_i}\mathtt{H}^{i,i+1}$, $j \in \{1,2\}$. Based on the relative homographies $\{\mathtt{H}^{i,i+1}\}$ we want to recover absolute homographies $\mathtt{H}^{v_i}$ that allow to represent all the matrices $\mathtt{P}^{j,\boldsymbol{\gamma}_i}$ in a common projective frame. In other terms, in the ideal case where there is no estimation error, the matrices $\mathtt{H}^{v_i}$ should satisfy

$$\mathtt{P}^{j,\boldsymbol{\gamma}_i}\,\mathtt{H}^{v_i} \cong \mathtt{P}^{j+i-1,*}, \qquad j \in \{1,2,3\}. \tag{10}$$

Obviously, the set $\{\mathtt{H}^{v_i}\}$ can only be determined up to an overall projective homography.

Proposition 2. *If for some $i = 1,\ldots,n$, the cameras $\mathtt{P}^{i+1,*}$ and $\mathtt{P}^{i+2,*}$ have different centers, then $\mathtt{H}^{v_i} \cong \mathtt{H}^{i,i+1}\mathtt{H}^{v_{i+1}}$. Furthermore, if the centers of each pair of consecutive cameras are different, then one can find a projective coordinate frame so that*

i) $\mathtt{H}^{v_i} = \mathtt{H}^{i,i+1}\mathtt{H}^{v_{i+1}}, \qquad \forall i = 1,\ldots,n-1,$

ii) $\alpha\,\mathtt{H}^{v_n} = \mathtt{H}^{n,1}\mathtt{H}^{v_1}$, *where α can be determined by* $\alpha = \frac{1}{4}\mathrm{Trace}(\prod_{i=1}^{n}\mathtt{H}^{i,i+1})$,

iii) Let $\bar{\mathtt{H}}$ be the $(4n)\times 4$ matrix resulting from the vertical concatenation of matrices $\mathtt{H}^{v_i}$. The four columns of $\bar{\mathtt{H}}$ are orthonormal.

This result, the proof of which is presented in the supplemental material, allows us to define the following algorithm for estimating the matrices $\{\mathtt{H}^{v_i}\}$. Given the relative homographies $\{\mathtt{H}^{i,i+1}\}$, we first compute α according to the formula in ii) and then minimize the cost function

$$\sum_{i=1}^{n-1}\frac{\|\mathtt{H}^{v_i} - \mathtt{H}^{i,i+1}\mathtt{H}^{v_{i+1}}\|_2^2}{\max(\sigma_{v_i}^2,\sigma_{v_{i+1}}^2)} + \frac{\|\alpha\,\mathtt{H}^{v_n} - \mathtt{H}^{n,1}\mathtt{H}^{v_1}\|_2^2}{\max(\sigma_{v_1}^2,\sigma_{v_n}^2)} \tag{11}$$

w.r.t. $\{\mathtt{H}^{v_i}\}$, subject to the orthonormality of the columns of $\bar{\mathtt{H}}$. Here, $\|\cdot\|_2$ is the Frobenius norm. The exact solution of this (non-convex) optimization problem can be computed using the singular value decomposition of a matrix of size $4n \times 4n$ constructed

First frame Second frame Last frame

Fig. 2. This figure illustrates the improvement achieved at each step of our algorithm. If the cameras are reconstructed without imposing loop constraints, the epipolar lines between the first and the last frames are extremely inaccurate (1st row). They become much more accurate when the constrained optimization is performed (2nd row). Finally, the result is almost perfect once the homography registration is done.

from α and $\{\mathtt{H}^{i,i+1}\}$. Since this is quite standard (based on the Courant-Fisher minimax theorem [30, Thm. 8.1.2]), we do not present more details here.

4.2 The Case of Several Loops

Assume now that we have identified several loops in the graph of trifocal tensors. Let N_{loop} be the number of these loops. We apply to each loop the method of the previous section and get a homography for every node of the loop. In general, one node of $\mathcal{G}_{\text{triplet}}$ may lie in several loops, in which case we will have several homographies for that node. It is then necessary to enforce the coherence of these homographies. To this end, we define the graph $\mathcal{G}_{\text{loop}}$ having N_{loop} nodes, each node representing a loop. Two nodes of $\mathcal{G}_{\text{loop}}$ are linked by an edge, if the corresponding loops have non-empty intersection. We will assume that the graph $\mathcal{G}_{\text{loop}}$ is connected, since otherwise it is impossible to simultaneously calibrate different connected components.

The next step consists in determining a minimal depth spanning tree $\mathcal{T}_{\text{loop}}$ of $\mathcal{G}_{\text{loop}}$. Since the number of loops is assumed small, this step will not be time consuming. Let $(\mathcal{L}, \mathcal{L}')$ be a pair of adjacent nodes of $\mathcal{T}_{\text{loop}}$. By an argument analogous to that of Proposition 2, one can show that there exists a 4×4 homography $\mathtt{H}^{\mathcal{L},\mathcal{L}'}$ such that $\mathtt{H}^{v,\mathcal{L}} \cong \mathtt{H}^{v,\mathcal{L}'}\mathtt{H}^{\mathcal{L}',\mathcal{L}}$ up to an estimation error, for every triplet of cameras $v \in \mathcal{L} \cap \mathcal{L}'$. Here, $\mathtt{H}^{v,\mathcal{L}}$ (resp. $\mathtt{H}^{v,\mathcal{L}'}$) stands for the homography assigned (*cf.* previous subsection) to the triplet v as a part of the loop $\mathcal{L}$ (resp. $\mathcal{L}'$). The homography $\mathtt{H}^{\mathcal{L}',\mathcal{L}}$ can be estimated by minimizing the objective function

$$\sum_{v \in \mathcal{L} \cap \mathcal{L}'} \|\alpha_v \mathtt{H}^{v,\mathcal{L}} - \mathtt{H}^{v,\mathcal{L}'} \mathtt{H}^{\mathcal{L}',\mathcal{L}}\|_2^2 / \sigma_v^2 \tag{12}$$

w.r.t. the matrix $\mathtt{H}^{\mathcal{L}',\mathcal{L}}$ and parameters $\{\alpha_v\}$ subject to $\|\mathtt{H}^{\mathcal{L}',\mathcal{L}}\|_2^2 + \sum_{v \in \mathcal{L} \cap \mathcal{L}'} \alpha_v^2 = 1$. Once again, this minimization can be carried out by computing the eigenvector corresponding to the smallest singular value of a suitably defined matrix.

Finally, to enforce the coherence of absolute homographies computed using different loops, we proceed as follows. We do not modify the homographies computed within the loop $\mathcal{L}_0$ constituting the root of the minimal depth spanning tree $\mathcal{T}_{\text{loop}}$. For any other loop $\mathcal{L}$, let $\mathcal{L}_0 \to \mathcal{L}_1 \to \ldots \to \mathcal{L}_k \to \mathcal{L}$ be the (unique) path joining $\mathcal{L}$ to the root. Then, every absolute homography $\mathtt{H}^{v,\mathcal{L}}$, $v \in \mathcal{L}$, computed within the loop $\mathcal{L}$ using the method of the previous subsection is replaced by $\mathtt{H}^{v,\mathcal{L}} \mathtt{H}^{\mathcal{L},\mathcal{L}_k} \cdots \mathtt{H}^{\mathcal{L}_1,\mathcal{L}_0}$. After this modification, the images by $\mathtt{H}^{v,\mathcal{L}}$ of the projection matrices $\mathtt{P}^{j,\gamma_v}$ $(j = 1, 2, 3)$ will all lie in nearly the same projective space. This makes it possible to recover the final projection matrices $\mathtt{P}^i$ by a simple computation presented in the next subsection.

4.3 Estimating Projection Matrices

Once the set of absolute homographies estimated, we turn to the estimation of camera matrices $\{\mathtt{P}^{j,*}\}$. Due to the estimates computed in previous steps, each projection matrix $\mathtt{P}^{j,*}$ can be estimated independently of the others. To ease notation and since there is no loss of generality, let us focus on the estimation of $\mathtt{P}^{1,*}$. We start by determining the nodes in $\mathcal{G}_{\text{triplet}}$ that contain the first view. Let $\mathcal{V}_1$ denote the set of these nodes. To each node $v \in \mathcal{V}_1$ corresponds one estimator of $\mathtt{P}^{1,*}$, denoted by $\mathtt{P}^{1,\gamma_v}$. Furthermore, we have a set of estimated homographies $\mathtt{H}^{v,\mathcal{L}}$ that satisfy, up to an estimation error, the relation $\mathtt{P}^{1,v}\mathtt{H}^{v,\mathcal{L}} \cong \mathtt{P}^{1,*}$. This is equivalent to $\alpha_{v,\mathcal{L}}\mathtt{P}^{1,v}\mathtt{H}^{v,\mathcal{L}} = \mathtt{P}^{1,*}$, $\forall v \in \mathcal{V}_1$, $\forall \mathcal{L} \supset \{v\}$ with some $\alpha_{v,\mathcal{L}} \in \mathbb{R}$. In these equations, the unknowns are the reals $\alpha_{v,\mathcal{L}}$ and the matrix $\mathtt{P}^{1,*}$. Since this matrix should be of rank 3, it has nonzero Frobenius norm. Therefore, we estimate $\mathtt{P}^{1,*}$ by $\mathtt{P}^1$ defined as a solution to

$$\arg\min_{\mathtt{P}} \min_{\{\alpha_{v,\mathcal{L}}\}:\|\mathtt{P}\|_2^2+\|\boldsymbol{\alpha}\|_2^2=1} \sum_{\mathcal{L}} \sum_{v \in \mathcal{L} \cap \mathcal{V}_1} \|\alpha_{v,\mathcal{L}}\mathtt{P}^{1,v}\mathtt{H}^{v,\mathcal{L}} - \mathtt{P}\|_2^2 / \sigma_v^2, \tag{13}$$

where $\boldsymbol{\alpha}$ stands for the vector having as coordinates the numbers $\alpha_{v,\mathcal{L}}$. Once again, the problem (13) can be explicitly solved using the SVD of an appropriate matrix.

5 Experiments

Implementation. In order to apply the methodology we have just described, we extract and match SIFT [31] descriptors from all the images. Then, epipolar geometries are estimated by DEGENSAC [32]. Note that some speed-up in this step can be achieved by using one of the recent versions of RANSAC [26,33]. Estimated EGs allow us to identify and remove wrong correspondences as well as to create feature tracks. Using these tracks and EGs as input for our algorithm, we compute as output the projection matrices of all the cameras. In order to be able to visually assess the reconstruction quality, all cameras and the 3D structure are upgraded to Euclidean [27].

Table 1. Characteristics of the datasets used for the experimental validation. From left to right: number of frames in each sequence, the resolution of each image, the number of 2D image points used for the final BA for our method and for bundler [5], the mean squared reprojection error.

Dataset	#frames	resolution	# image points		MSRE (pxl)	
			Our	Bundler	Our	Bundler
Dinosaur	36	576 × 720	45,250	37,860	0.27	0.25
Temple	45	480 × 500	26,535	23,761	0.08	0.11
Fountain P11	11	2048 × 3072	57,547	23,648	0.16	0.13
Herz-Jesu R23	23	2048 × 3072	129,803	–	0.41	–
Detenice	34	1536 × 2048	30,200	–	0.15	–
Calvary	52	2624 × 3972	54,798	–	0.51	–

Fig. 3. One frame of each dataset used to test our methodology. From left to right: dinosaur, temple, fountain P11, Herz-Jesu R23 [34], Calvary, Detenice fountain.

Datasets. We tested our methodology on six datasets: the *dinosaur* sequence (36 frames), the *temple* sequence (45 frames), the *fountain P11* sequence (11 frames), the *Herz-Jesu R23* sequence (23 frames), the *Detenice fountain* sequence (34 frames) and the *calvary* sequence (52 frames). For the first three datasets, the ground truth of camera matrices is available on the Web.

Quality measures. Since the main contribution of the present paper concerns the projective reconstruction, it is natural to assess the quality of the proposed approach using the distance:

$$d_{proj}(\{\mathtt{P}^j\},\{\mathtt{P}^{j,*}\}) = \inf_{\boldsymbol{\alpha},\mathtt{H}} \sum_{j=1}^{n} \|\alpha_j \mathtt{P}^j \mathtt{H} - \mathtt{P}^{j,*}\|_2^2, \tag{14}$$

where $\mathtt{P}^j$ and $\mathtt{P}^{j,*}$ are respectively the reconstructed and the true camera projection matrices, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ is a vector of real numbers and $\mathtt{H}$ is a 3D-homography. Naturally, this measure can be used only on sequences for which the ground truth is available. Note also that the computation of the infimum in (14) is a non-convex optimization problem. We solve it by first computing the one-norm solution to the least squares problem $\min_{\boldsymbol{\alpha},\mathtt{H}} \sum_{j=1}^{n} \|\mathtt{P}^j\mathtt{H} - \alpha_j^{-1}\mathtt{P}^{j,*}\|_2^2$, and then use this solution as a starting point for an alternating minimization. For the examples considered here, this converges very rapidly and, since the results are good, we believe that the local minimum we find is in fact a global minimum, or at least not too far from it.

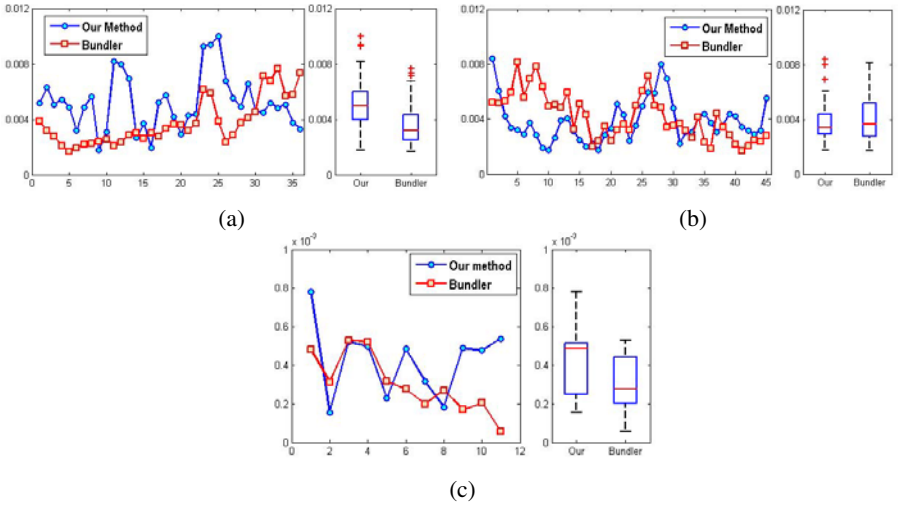


Fig. 4. This figure shows the errors in estimated camera matrices for our method and for bundler. The per-camera errors and their boxplots for the dinosaur sequence (a), the temple sequence (b) and for the fountain P11 sequence (c). One can remark that our method achieves the same level of accuracy as that of bundler, despite the fact that we do not use any information on the internal parameters, while bundler assumes that the skew is zero and the principal point is the center.

Results. For the dinosaur, temple and fountain P11 sequences, since ground truth exists, we compared our results with those of bundler [5], which is a state-of-the-art calibration software. The ground truth was normalized so that the Frobenius norm of all the cameras is one. For both reconstructions (ours and bundler), we computed numbers α_j and a homography $\mathtt{H}$ by minimizing (14). This allows us to define the per-camera error as $\|\alpha_j \mathtt{P}^j \mathtt{H} - \mathtt{P}^{j,*}\|_2^2$ for the jth camera. As shown in Fig. 4, not only these errors are small, but also our results are quite comparable to those of bundler despite the fact that our method does not perform intermediate BAs and does not assume that the principal point is in the center and the skew is zero. One can also note that the error is well distributed over the whole sequence of cameras due to the fact that both methods operate on the closed sequence. Furthermore, the results reported for fountain P11 are achieved without final BA, proving that the method we proposed furnishes a good starting point for the non-linear optimization.

As for the datasets where no ground truth is known, we have chosen to use as measure of evaluation the multiview stereo reconstruction of the scene based on the method of [25].[2] The results are shown in Fig. 1 (right) for the calvary sequence and in Fig. 5 for the Herz-Jesu R23 and the Detenice fountain sequences. In the aim of comparing our results with other approaches, let us recall that (as reported in [34]) on the Herz-Jesu R23 data the ARC3D software succeeded to calibrate four of the 23 cameras, while the method proposed in [4] calibrated all the cameras with a relatively large error for

[2] Since multiview stereo reconstruction is not the purpose of the paper and is only used for illustration, the results shown in Fig. 1 and 5 are obtained without the final mesh refinement.

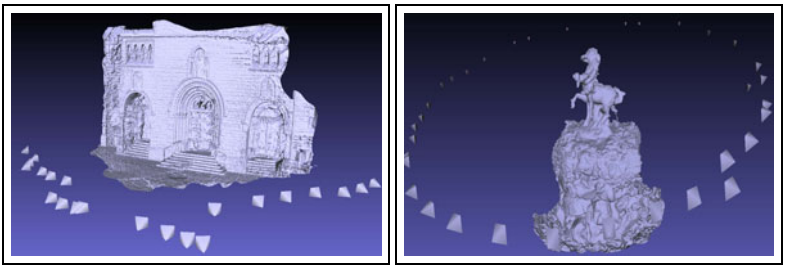


Fig. 5. Multi-view stereo reconstruction using the camera matrices estimated by our method for the Herz-Jesu R23 and Detenice fountain datasets. For these data, the ground truth is unavailable but the quality of the scene reconstruction demonstrates the accuracy of estimated cameras.

cameras 6-11. Although we are unable to quantitatively compare our reconstruction to that of [4], the accuracy of the 3D scene reconstruction makes us believe that the estimated cameras are very close to the true ones.

6 Conclusion

In this paper, we have proposed a new approach to the problem of autocalibration of a network of cameras. Our approach is based on a representation of the network of cameras by a graph of trifocal tensors and on a natural parameterization of camera matrices and relating homographies. We have proposed to estimate the unknown parameters by a constrained optimization that can be recast in a linear program. Thanks to the sparsity of the matrices involved in this linear program, the running times of the proposed algorithm are very attractive even for large scale datasets. The experiments reported in this paper show that our approach leads to state-of-the-art results without assuming any kind of information on the internal parameters.

Acknowledgments. We thank Vu Hiep for the results of multi-view stereo experiments, Daniel Martinec for the fountain dataset, Imagine for the calvary dataset and Christoph Strecha for fountain P11 and Herz-Jesu R23 datasets. This work was partially supported by ANR under grant Callisto.

References

1. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision, 2nd edn. Cambridge University Press, Cambridge (2003)
2. Faugeras, O., Luong, Q.T., Papadopoulou, T.: The Geometry of Multiple Images: The Laws That Govern The Formation of Images of A Scene and Some of Their Applications. MIT Press, Cambridge (2001)
3. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3D. ACM Press, New York (2006)
4. Martinec, D., Pajdla, T.: Robust rotation and translation estimation in multiview reconstruction. In: CVPR (2007)

5. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from Internet photo collections. Int. J. Comput. Vision 80, 189–210 (2008)
6. Furukawa, Y., Ponce, J.: Accurate camera calibration from multi-view stereo and bundle adjustment. International Journal of Computer Vision 84, 257–268 (2009)
7. Bujnak, M., Kukelova, Z., Pajdla, T.: 3D reconstruction from image collections with a single known focal length. In: ICCV, pp. 351–358 (2009)
8. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building rome in a day. In: ICCV (2009)
9. Havlena, M., Torii, A., Knopp, J., Pajdla, T.: Randomized structure from motion based on atomic 3d models from camera triplets. In: CVPR (2009)
10. Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., Sayd, P.: Generic and real-time sfm using local bundle adjustment. Image Vision Comput 27, 1178–1193 (2009)
11. Fitzgibbon, A.W., Zisserman, A.: Automatic camera recovery for closed or open image sequences. In: Burkhardt, H.-J., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1406, pp. 311–326. Springer, Heidelberg (1998)
12. Avidan, S., Shashua, A.: Threading fundamental matrices. IEEE Trans. Pattern Anal. Mach. Intell. 23, 73–77 (2001)
13. Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. Int. J. Comput. Vision 59, 207–232 (2004)
14. Sinha, S.N., Pollefeys, M., McMillan, L.: Camera network calibration from dynamic silhouettes. In: CVPR (2004)
15. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. Int. J. Comput. Vision 9, 137–154 (1992)
16. Sturm, P., Triggs, B.: A factorization based algorithm for multi-image projective structure and motion. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065, pp. 709–720. Springer, Heidelberg (1996)
17. Jacobs, D.: Linear fitting with missing data: Applications to structure-from-motion and to characterizing intensity images. In: CVPR, p. 206 (1997)
18. Martinec, D., Pajdla, T.: 3D reconstruction by fitting low-rank matrices with missing data. In: CVPR, pp. I: 198–205 (2005)
19. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment - a modern synthesis. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) ICCV-WS 1999. LNCS, vol. 1883, pp. 298–372. Springer, Heidelberg (2000)
20. Klopschitz, M., Zach, C., Irschara, A., Schmalstieg, D.: Generalized detection and merging of loop closures for video sequences. In: 3DPVT (2008)
21. Scaramuzza, D., Fraundorfer, F., Pollefeys, M.: Closing the loop in appearance-guided omnidirectional visual odometry by using vocabulary trees. Robot. Auton. Syst. (to appear, 2010)
22. Cornelis, N., Cornelis, K., Van Gool, L.: Fast compact city modeling for navigation previsualization. In: CVPR (2006)
23. Tardif, J., Pavlidis, Y., Daniilidis, K.: Monocular visual odometry in urban environments using an omnidirectional camera. In: IROS, pp. 2531–2538 (2008)
24. Torii, A., Havlena, M., Pajdla, T.: From google street view to 3d city models. In: OMNIVIS (2009)
25. Vu, H., Keriven, R., Labatut, P., Pons, J.P.: Towards high-resolution large-scale multi-view stereo. In: CVPR (2009)
26. Chum, O., Matas, J.: Matching with PROSAC: Progressive sample consensus. In: CVPR, vol. I, pp. 220–226 (2005)
27. Ponce, J., McHenry, K., Papadopoulo, T., Teillaud, M., Triggs, B.: On the absolute quadratic complex and its application to autocalibration. In: CVPR, pp. 780–787 (2005)
28. Quan, L.: Invariants of six points and projective reconstruction from three uncalibrated images. IEEE Trans. Pattern Anal. Mach. Intell. 17, 34–46 (1995)

29. Schaffalitzky, F., Zisserman, A., Hartley, R.I., Torr, P.H.S.: A six point solution for structure and motion. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1842, pp. 632–648. Springer, Heidelberg (2000)
30. Golub, G.H., Van Loan, C.F.: Matrix computations, 3rd edn. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore (1996)
31. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
32. Chum, O., Werner, T., Matas, J.: Two-view geometry estimation unaffected by a dominant plane. In: CVPR, vol. I, pp. 772–779 (2005)
33. Sattler, T., Leibe, B., Kobbelt, L.: Scramsac: Improving ransac‘s efficiency with a spatial consistency filter. In: ICCV (2009)
34. Strecha, C., von Hansen, W., Van Gool, L., Fua, P., Thoennessen, U.: On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery. In: CVPR (2008)

Efficient Structure from Motion by Graph Optimization

Michal Havlena[1], Akihiko Torii[1,2], and Tomáš Pajdla[1]

[1] Center for Machine Perception, Department of Cybernetics, Faculty of Elec. Eng., Czech Technical University in Prague, Technická 2, 166 27 Prague 6, Czech Republic
{havlem1,pajdla}@cmp.felk.cvut.cz
[2] Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo, Japan
torii@ctrl.titech.ac.jp

Abstract. We present an efficient structure from motion algorithm that can deal with large image collections in a fraction of time and effort of previous approaches while providing comparable quality of the scene and camera reconstruction. First, we employ fast image indexing using large image vocabularies to measure visual overlap of images without running actual image matching. Then, we select a small subset from the set of input images by computing its approximate minimal connected dominating set by a fast polynomial algorithm. Finally, we use task prioritization to avoid spending too much time in a few difficult matching problems instead of exploring other easier options. Thus we avoid wasting time on image pairs with low chance of success and avoid matching of highly redundant images of landmarks. We present results for several challenging sets of thousands of perspective as well as omnidirectional images.

Keywords: Structure from motion, Image set reduction, Task prioritization, Omnidirectional vision.

1 Introduction

We seek to reconstruct 3D scene structure and camera poses from a large collection of images downloaded from the web or taken by a camera mounted on a moving vehicle as in the Google Street View. This is a challenging task because unstructured web collections often contain a large number of very similar images of landmarks while, on the other hand, image sequences often have very limited overlap between images. Computation effort of large scale structure from motion is dominated by image matching, which is often done only to find that matched images actually do not have visual overlap.

Most of the state-of-the-art techniques for 3D reconstruction from unorganized image sets [1,2,3,4] start the computation by performing exhaustive pairwise image matching which becomes infeasible for image sets comprising thousands of images. Even Photo Tourism [5], one of the most known 3D modeling systems from unordered image sets, uses exhaustive pairwise image feature matching and exhaustive pairwise epipolar geometry computation to create the image graph

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 100–113, 2010.

with vertices being images and edges weighted by the uncertainty of pairwise relative position estimations which is later used to lead the reconstruction. By finding the skeletal set [6] as a subgraph of the image graph having as few internal nodes as possible while keeping a high number of leaves and the shortest paths being at most constant times longer, the reconstruction time improves significantly but the time spent on image matching remains the same. Recent advancement of the aforementioned technique [7] abandons exhaustive pairwise image matching by using shared occurrences of visual words [8,9] to match only the ten most promising images per each input image. On the other hand, the number of computed image matchings still remains rather high for huge image sets. The presented computational speed is achieved also thanks to massive parallelization which demands grid computing on 496 cores.

We aim at reducing the number of image matchings by reducing the size of the image set, because it may be highly redundant. Opposed to the technique presented in [10], we do not cluster the input images using GIST [11] but we select a subset of input images in such a way that all the remaining images have a significant visual overlap with at least one image from the selected ones (Section 2). As this visual overlap is measured by shared occurrences of visual words [9], the method is more robust to viewpoint changes because it seeks for images capturing the same 3D structure rather than for images acquired from the same viewpoint, as demonstrated in [12]. Furthermore, the method works also for omnidirectional images where GIST often fails. For selecting the subset of input images, the approximate minimal connected dominating set is computed by a fast polynomial algorithm [13] on the graph constructed according to the visual overlap. The algorithm used is closely related to the maximum leaf spanning tree algorithm employed in [6] but the composition of the graph is quite different and less computationally demanding in our case.

The actual SfM pipeline uses the atomic 3D models reconstructed from camera triplets introduced by [14] as the basic elements of the reconstruction but the strict division of the computation into steps is relaxed by introducing a priority queue which interleaves different reconstruction tasks in order to get a good scene covering reconstruction in limited time (Section 3). Our aim here is to avoid spending too much time in a few difficult matching problems by exploring other easier options which lead to a comparable resulting 3D model in shorter computational time. We also introduce model growing by constructing new 3D points when connecting an image which allows for sparser image sets than those which could be reconstructed by [14].

2 Image Set Reduction

When performing sparse 3D reconstruction from user-input images, the input image set may often be highly redundant, such as photographs acquired by tourists at landmark sites. As it is not needed to use all such input images in order to get a 3D model covering the scene captured in them, it is possible to speed the reconstruction up by using only a suitable subset of input images.

Algorithm 1. Approximate minimum CDS computation [13]

Input Unweighted undirected graph $G = (V, E)$.
Output List S of vertices belonging to the minimum CDS of G.

I. Label all vertices $v \in V$ white.
II. Set $D := \{\}$ and repeat until no white vertices are left:
  1: For all black vertices $v \in V$ set $c(v) := 0$.
  2: For all gray and white vertices v set $c(v) :=$ number of white neighbours of v.
  3: Set $v^* := \arg\max_v c(v)$.
  4: Label v^* black and add it into D.
  5: Label all neighbours of v^* gray.
III. Set $S := D$ and connect components of the subgraph of G induced by D by adding at most 2 vertices per component into S in a greedy way.
IV. Return S.

We seek for a method that would remove the unnecessary images from the input image set while affecting neither the quality nor the connectivity of the resulting 3D model much. The concept of visual words, which first appeared in [9], has been used successfully for matching images and scenes [8]. It proved its usefulness also for near duplicate image detection [12] when the scene is captured from different viewpoints or under different lighting conditions. Our aim is to (i) evaluate pairwise image similarity efficiently following [15,7] and (ii) formulate the selection of the desired subset of input images as finding a suitable subgraph of the graph constructed according to image similarity.

2.1 Image Similarity

We use the bag-of-words approach to evaluate image similarity. In particular, we follow the method proposed in [15] to create the pairwise image similarity matrix M_{II} containing the cosines of the angles between the normalized tf-idf vectors computed from the numbers of occurrences of the quantized SURF [16] image feature descriptors in individual images. Next, we create an unweighted undirected graph G_{II} expressing image similarity. Vertices of G_{II} are the input images and we add five edges per vertex connecting it with the five most similar images according to the values of M_{II}, which is close to the approach used in [7]. Edges are not added if the measured similarity falls under 0.05. Notice that there may (and often will) exist vertices with degree higher than five in the resulting graph as some images may be similar to many other images.

2.2 Minimum Connected Dominating Set

According to [13], the minimum connected dominating set (CDS) problem is defined as follows. Given a graph $G = (V, E)$, find a minimum size subset S of vertices, such that the subgraph induced by S is connected and S forms a

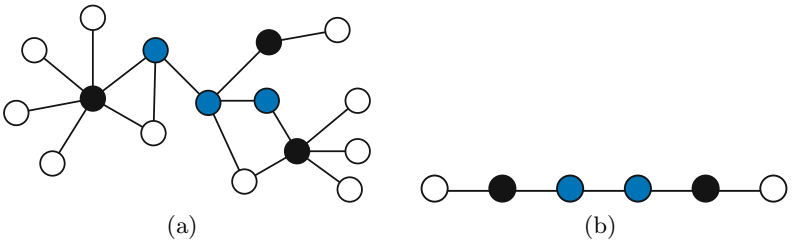


Fig. 1. Minimum CDS computation. Vertices belonging to the minimum dominating set D are labeled black, vertices added when connecting the components in order to get S are labeled blue. (a) General graph. (b) Graph being a singly connected line.

dominating set in G. In a graph with a dominating set, each vertex is either in the dominating set or adjacent to some vertex in the dominating set. The problem of finding the minimum CDS is known to be *NP*-hard [17] but [13] presents a fast polynomial algorithm with an approximation ratio of $\ln \triangle + 3$, $\triangle$ being the maximum vertex degree in the graph, see Algorithm 1.

We use the aforementioned algorithm to find the minimum connected dominating set S_{II} of the graph G_{II}, see Figure 1(a), and *only the images corresponding to the vertices in S_{II} are further used for the sparse 3D model reconstruction.* Edges of the subgraph of G_{II} induced by D (Algorithm 1, Step III.) together with the edges connecting the components of this subgraph in order to get S_{II} are used as the seeds of the reconstruction.

The usage of the dominating set provides for connecting the removed images to the resulting 3D model reconstructed from the selected ones using camera resectioning [18] if required, as an image is removed only if it is similar to at least one image which remains in the selected subset, i.e. there exists visual overlap between the resulting model and each of the removed images. Furthermore, the connectivity of the resulting 3D model is preserved by using the connected dominating set which does not allow for splitting the originally connected graph into components. For non-redundant image sets, e.g. when the graph expressing image similarity is a singly connected line, the method removes only the first and the very final images because removing more images would affect model connectivity, see Figure 1(b). On the other hand, the reduction of highly redundant image sets is drastic, as shown in Section 4.1.

3 3D Model Construction Using Tasks Ordered by a Priority Queue

The reduced image set is input into our 3D reconstruction pipeline which grows the resulting 3D model from several atomic 3D models. The computation is divided into tasks, each of them can either try to create a new atomic 3D model from three images, or try to connect one image to a given 3D model, see Figure 2. The order of the execution of different tasks is determined by task priority

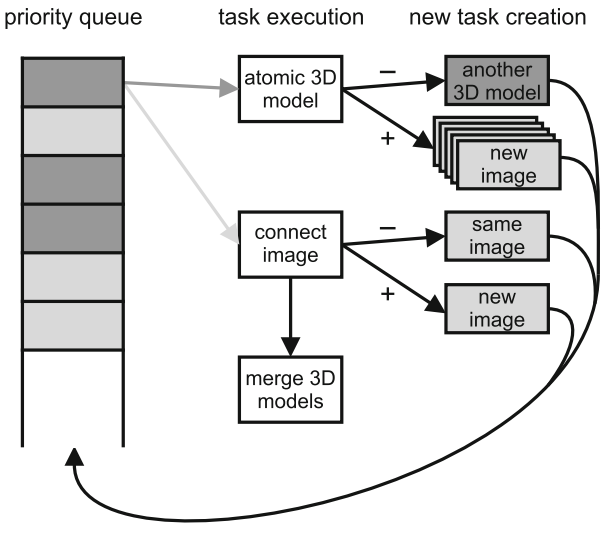


Fig. 2. Schematic visualization of the computation. The task retrieved from the head of the priority queue can be either an atomic 3D model construction task (dark gray) or an image connection task (light gray). Unsuccessful atomic 3D model reconstruction (–) inserts another atomic 3D model reconstruction task with the priority key doubled into the queue, a successful one (+) inserts five image connection tasks. Unsuccessful image connection (–) inserts the same task again with the priority key doubled, a successful one (+) inserts a new image connection task. Merging of overlapping 3D models is called implicitly after every successful image connection if the overlap is sufficient.

keys set when adding them to the priority queue being the essential underlying data structure. Note that *the task with the smallest priority key has the highest priority*, i.e. it is always in the head of the queue, in our implementation of the priority queue. Our aim is to set task priority keys in such a way that stopping the computation at any time would give a good scene covering sparse 3D model for the time given which is demanded e.g. by online SfM services. The state-of-the-art SfM approaches [5,6,7] implement this priority queue implicitly in such a way that they may get stuck by solving a difficult part of the reconstruction even when an easier path to the goal exists, as they are greedily growing from a single seed. Using our approach, several seeds are grown in parallel so the easiest path is actively searched for.

First, the queue is filled with one candidate camera triplet for atomic 3D model reconstruction per seed. The triplet is constructed from the two cameras C_1, C_2 being the endpoints of the edge corresponding to the seed. The third camera C_3^* is selected as

$$C_3^* = \arg\max_{C_3} \min(M_{II}(C_1, C_3), M_{II}(C_2, C_3)) \quad (1)$$

and the priority key of this task is set to $1 - M_{II}(C_1, C_2)$.

Next, the task from the head of the priority queue is taken and executed. As we are just starting the computation, it will be an atomic 3D model creation task. If the atomic 3D model reconstruction from a given candidate camera triplet is not successful, the camera triplet is rejected and another candidate camera triplet for the same seed is input into the queue with the priority key doubled. The new third camera accompanying cameras C_1 and C_2 is selected similarly as in Equation 1 by taking the camera C_3^* with the n-th largest value of $\min(M_{II}(C_1, C_3^*), M_{II}(C_2, C_3^*))$ and increasing n. After a successful atomic 3D model creation, the vicinity of the respective seed is searched for camera candidates suitable for connecting with the newly created atomic 3D model and tasks connecting the five most suitable cameras are input into the queue. We put the cameras contained in the atomic 3D model into the set $\mathcal{C}^c$ and the rest of the cameras into $\mathcal{C}^n$. Then, we search for a candidate camera C_r^* to be connected to the atomic 3D model using

$$(C_r^*, C_s^*) = \underset{(C_r, C_s) \in \mathcal{C}^n \times \mathcal{C}^c}{\arg\max} M_{II}(C_r, C_s). \tag{2}$$

The priority key of this task is set to $1 - M_{II}(C_r^*, C_s^*)$. Other four candidate cameras are selected similarly using the second, third, fourth, and fifth largest value of $M_{II}(C_r^*, C_s^*)$.

Alternatively, the head of the priority queue may contain an image connection task. After a successful image connection, a task connecting another camera to the same partial 3D model is created using Equation 2 again with a larger set $\mathcal{C}^c$ and input into the queue in order to keep the number of image connection tasks at five per a partial model. When the connection of an image to a given 3D model is unsuccessful, the task is input into the queue again with the priority key doubled because it may be successful if tried again after other images are successfully connected. In order to keep the resulting reconstruction consistent and connected, grown 3D models are implicitly merged together when they share at least five images. If the merge is not successful, it will be tried again when the number of shared images increases again.

The whole procedure is repeated until the priority queue is empty or the available time runs out. The following paragraphs describe particular parts of the pipeline in deeper detail.

3.1 Creation of Atomic 3D Models

Atomic 3D model reconstruction introduced in [14] has been improved and extended in several ways:

1. SIFT [19] and SURF [16] image feature detectors and descriptors have been added as it shows out that a combination of many different detectors is needed for difficult image sets. On the other hand, for easy image sets, it is possible to use only the fastest of them, which is SURF in our case.
2. Camera calibration does not need to be the same for all images in the set and can be obtained from the EXIF info of JPEG images.

3. The formula computing the quality score q has been simplified into:

$$q = |\{X : \tau(X) \geq \ 5^\circ\}|, \tag{3}$$

$\tau(X)$ being the apical angle measured at the 3D point X. In contrast with the original formula, 3D points with even larger apical angles do not contribute more to the quality score as we found out that it does not bring any significant improvement over the simple formula.

We require the quality score of at least 20 to accept a given candidate camera triplet as being suitable for reconstructing. Together with the remaining triplet quality pre-tests, the decision rule is the following: A given candidate camera triplet is accepted if and only if the results of pairwise epipolar geometries are consistent (the inlier ratio of the RANSAC finding the common scale is higher than 0.7), at least fifty 3D points have been reconstructed, at least twenty of them have apical angles larger than 5 degrees, and their projections cover a sufficiently large portion of the three respective viewfields.

3.2 Model Growing by Connecting Images

Connection of a new image to a given partial 3D model proceeds in two stages. First, the pose of the corresponding camera C_g with respect to the 3D model is estimated. Secondly, promising cameras from the vicinity of the newly connected one are used to create new 3D points.

Every 3D point already contained in the model has a descriptor which is transferred from one of the corresponding images during its triangulation. Thus it is easy to find 2D-3D matches between the reconstructed 3D points and the feature points detected and described in the candidate image being connected. To ensure reasonable speed even for large models with millions of points, we do one-way matching only with strict criteria on the first/second nearest neighbour distance ratio, setting it to 0.7 [19]. If the number of tentative matches is smaller than 20, the connection is not successful. Otherwise, RANSAC sampling triplets of 2D-3D matches is used to find the camera pose [18] having the largest support evaluated by the cone test [14]. Local optimization is achieved by repeated camera pose computation from all inliers [20] via SDP and SeDuMi [21]. We require the inlier ratio to be higher than 60% to consider the connection as successful and continue.

Next, we find the cameras already contained in the partial model, which have some viewfield overlap with the newly connected camera, by examining the projections of the inlier 3D points from the previous stage. We take a set $\mathcal{C}^p$ of all cameras, which contain projections of at least 20 inlier 3D points, and try to triangulate 3D points from camera pairs $(C_g, C_i) : C_i \in \mathcal{C}^p$. Newly triangulated 3D points with apical angles larger than 5 degrees are accepted if they are projected to at least three cameras after being merged based on the shared 2D feature points in C_g. Cone test can further reject a 3D point if those projections are not consistent with any possible 3D point position. Finally, sparse bundle adjustment [22] is used to refine the whole partial reconstruction after adding new 3D points and their projections.

3.3 Merging Overlapping Partial Models

When two partial 3D models share images, they usually share also 2D feature points which are the projections of some already triangulated 3D points. Therefore, we can avoid costly descriptor matching and create tentative 3D point matches between the two partial 3D models from pairs of 3D points which project to the same 2D feature points in both models.

As the 2D-3D matching used when connecting new images is rather strict, it often fails to find correspondences between not so distinctive regions, e.g. regions corresponding to the repetitive scene structures, which leads into triangulating the same scene 3D point once more at the latter stage. After connecting many images, scene 3D points may have several triangulated copies in the model, that is why the tentative 3D point matches created for merging often form large connected components, each of them corresponding to a single scene 3D point. After splitting all of these components into two parts, one per each partial model being merged, we use the cone test for each of those parts to verify that given 3D points can be merged into one. When this "internal merge" consolidating the partial models is finished, we continue with merging the two models using the collapsed tentative 3D point matches.

If there are less than 10 tentative 3D point matches, the merge is not successful, otherwise we try to find a similarity transform between the coordinate systems of the models. As three 3D point matches are needed to compute the similarity transform parameters [23], RANSAC with samples of length three is used. Inliers are evaluated by the cone test using image projections from both partial models and local optimization is performed by repeating the similarity transform computation from all inliers. Camera poses corresponding to the images shared by the models are averaged (rotation and position separately) inside the RANSAC loop before the cone test, so the similarity transforms which would lead into incorrectly averaged cameras would not be accepted. We require the inlier ratio to be higher than 60% to consider the merge as successful.

Finally, the smaller model is transformed to the coordinate system of the larger one because transforming the smaller model is faster. 3D point matches which were inliers are merged into a single point with the position being the mean of the former positions after transformation and duplicate image projections are removed. Sparse bundle adjustment [22] is used to refine the whole partial reconstruction after a successful merge.

4 Experiments

We demonstrate the proposed method in three experiments. The first one shows the efficient reduction of a highly redundant image set using the approximate minimum connected dominating set of a graph constructed using the image similarity matrix, the latter ones present the output of our 3D model reconstruction pipeline after 6 hours of computation for an omnidirectional and a perspective image set. All measured times are achieved by running a MATLAB+MEX implementation on a 2.83GHz Core2Quad PC.

4.1 Image Set Reduction

Image set DiTrevi consists of 2,545 images resulting from a Flickr Photo Sharing site [24] search for "di trevi" (April 2009). The image set is highly redundant and contaminated with images not capturing Di Trevi Fountain as it comprises pictures uploaded by hundreds of tourists visiting Rome. After detecting SURF image features and computing the image similarity matrix in 2 hours, the algorithm finding the approximate minimum connected dominating set of the corresponding graph returned 70 images in 5 seconds, see Figure 3. Selected images reasonably cover different scene viewpoints while the image set size was reduced by more than 97%. Furthermore, the contamination ratio of the image set decreased from 17% to 7% after the reduction.

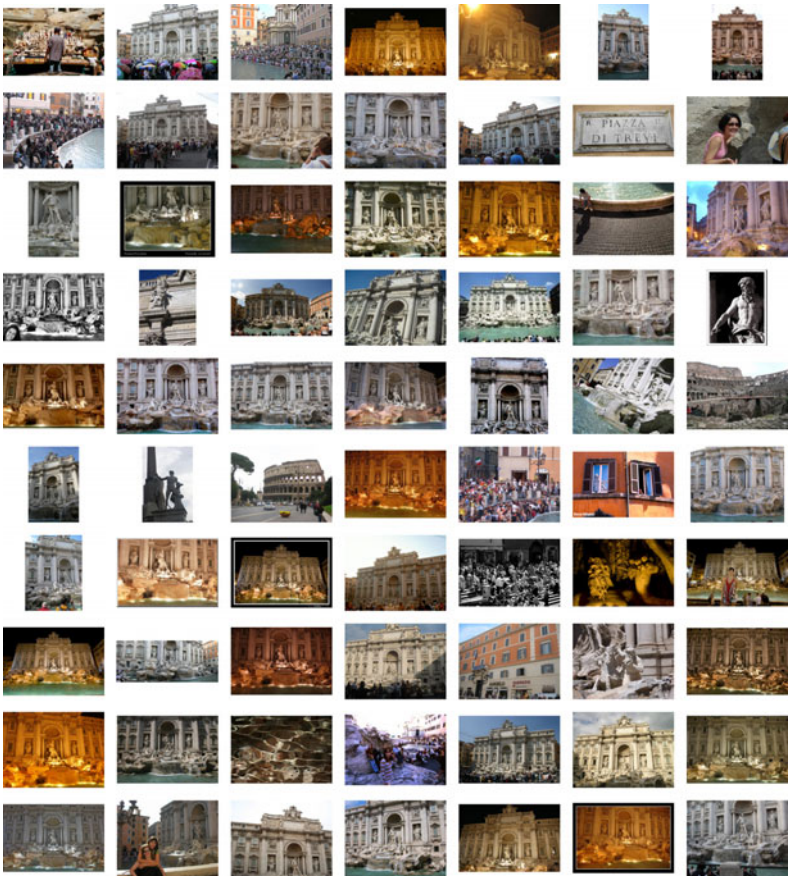
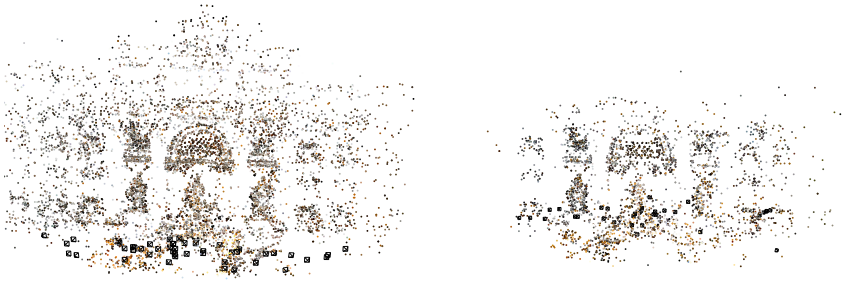


Fig. 3. Images corresponding to the approximate minimum connected dominating set computed for image set DiTrevi. Image set size has been reduced by 97% from 2,545 to 70 and the contamination ratio of the image set decreased from 17% to 7%.



(a) (b)

Fig. 4. (a) 3D model computed by Bundler [5] from the 70 images selected from image set DiTrevi by CDS. (b) The best from the 3D models returned by the five runs of Bundler on different random selections of 70 images from image set DiTrevi.

We used Bundler [5], a publicly available SfM tool, to evaluate the suitability of the image selection done by CDS for 3D reconstruction. The model returned in 44 minutes contains 47 camera poses and 8,489 3D points, see Figure 4(a). We ran Bundler also on five randomly selected sets of 70 images out of 2,545. Two of the runs did not return any result, two returned small fragments of the model with fewer than 5 camera poses, and one returned an incomplete 3D model having 32 camera poses and 3,355 3D points, as can be seen in Figure 4(b).

4.2 Sparse 3D Model Reconstruction

Two city sequences with landmark areas visited several times are used to demonstrate sparse 3D model reconstruction, see Figure 5. Nevertheless, they were input into the pipeline as unordered image sets.

Castle image set. Omnidirectional image set Castle [14] captured by a 180° fisheye lens camera with known calibration [25] consists of 4,472 omnidirectional images captured while walking in the center of Prague and around the Prague Castle. The obtained approximate minimum connected dominating set comprises 1,063 vertices and 1,359 edges are used as the seeds of the reconstruction. Image set reduction is not as drastic as for image set DiTrevi because the images are more evenly distributed. We use MSER [26], SIFT, and SURF image features in order to create sufficiently many 3D points even when image resolution is low. Several 3D models showing the important landmarks captured in the image set were obtained when the reconstruction time was limited to 6 hours, see Figure 6.

The resulting sparse 3D models are very similar to those presented in [14] but the speed of the reconstruction differs significantly as the authors of the aforementioned paper needed 12.5 days to obtain those results. Using our approach, the models are obtained in 10 hours, including 4 hours for image similarity matrix computation, which shows proper task priority key assignment.

Vienna image set. Image set Vienna [27] consists of 2,448 radially undistorted perspective images captured by a pre-calibrated camera while walking in the

Fig. 5. Sample input images from image sets Castle and Vienna respectively

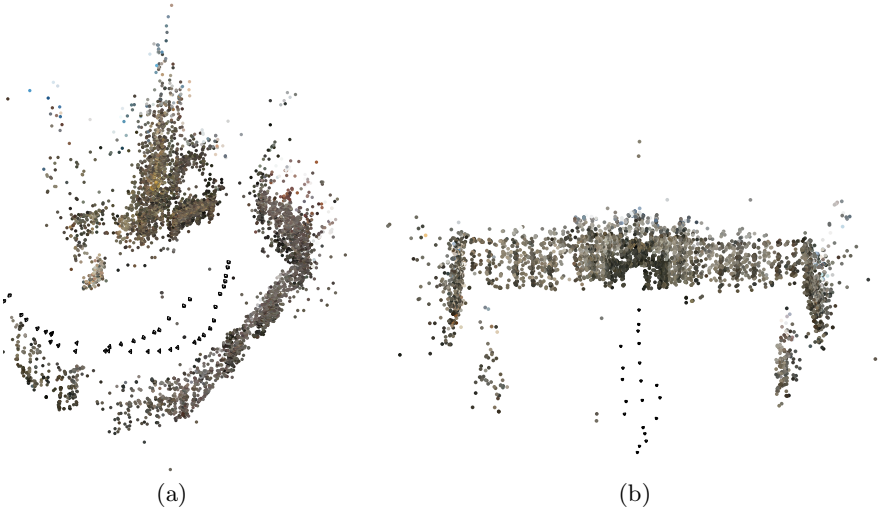
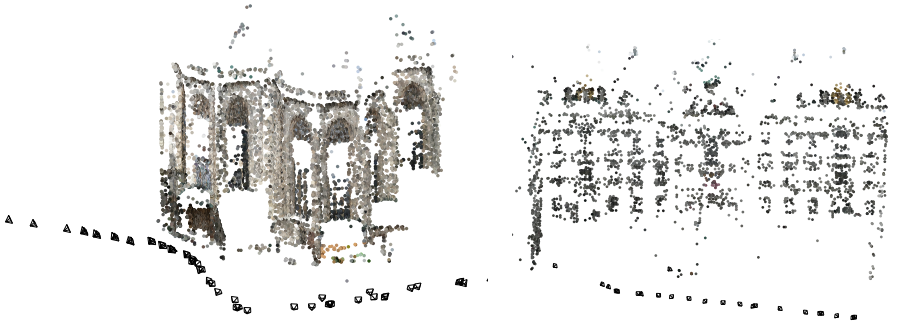


Fig. 6. Two largest partial 3D models reconstructed from the reduced image set Castle (1,063 images) after 6 hours of computation

center of Vienna. After computing the image similarity matrix in 90 minutes, 1,008 vertices and 1,900 edges being the seeds of the reconstruction are obtained in 10 seconds as the result of the search for the approximate minimum connected



(a) (b)

Fig. 7. Two largest partial 3D models reconstructed from the reduced image set Vienna (1,008 images) after 6 hours of computation

Table 1. The number of computed pairwise matchings, the number of active seeds, and the number of images contained in at least one partial model for the reduced image set Vienna (1,008 images) at given times of the reconstruction process

Time	1h	2h	3h	4h	5h	6h	7h	8h	9h	10h	11h	12h
# pairs	548	991	1432	1773	2100	2360	2624	2882	3172	3437	3679	4030
# seeds	44	57	66	77	86	80	79	77	73	71	71	65
# images	153	244	313	368	411	438	466	496	521	546	572	600

dominating set of the corresponding graph. As image resolution is sufficient, only SURF image features are used for 3D model reconstruction. The 3D models showing several important landmarks captured in the image set, received after 6 hours of reconstruction, can be seen in Figure 7.

Compared to the omnidirectional image set Castle, only parts of the landmarks are reconstructed in the 6 hour limit because more images are needed to capture the whole landmark as the field of view of the perspective camera is limited. Partial 3D models become larger and connected gradually when the reconstruction continues, see Table 1 for different quantitative results of the reconstruction process at given times. Notice that the number of active seeds drops ($86 \rightarrow 77 \rightarrow 65$) after some time as the overlapping models are merged and also the sub-quadratic number of computed pairwise matchings w.r.t. the number of images contained in the partial models being far behind the quadratic number which would be achieved by methods using exhaustive pairwise image matching.

Note that when running Bundler on the reduced image set, 3 hours are spent on detecting and describing SIFT image features and 1,922 out of 15,753 tested image pairs are accepted after additional 6 hours of computation. No partial 3D models are output at this time as bundling starts later, after all 507,528 possible image pairs are tested.

If one modified Bundler according to [7] so that it would test only the ten most promising image pairs per image based on image similarity and ran it on the non-reduced image set comprising 2,448 images, the whole 6 hour limit would still be spent on testing 16,762 obtained image pairs. This demonstrates the need for a prioritized structure from motion pipeline for large image sets.

5 Conclusions

We presented a pipeline for efficient sparse 3D model reconstruction from highly redundant unordered image sets, such as those acquired by tourists at landmark sites as well as image sequences. The approximate minimum connected dominating set of a graph constructed according to the image similarity matrix computed from tf-idf vectors over SURF image features is used both for (i) reducing the size of the image set by removing nearly duplicate images and (ii) setting priority keys of the reconstruction tasks stored in a priority queue. The proposed interlacing of different reconstruction tasks allows for obtaining either a good scene covering sparse 3D model in limited time or a complete sparse 3D model when time is not limited.

Based on our experiments, image similarity works very well for the presented image sets and the number of the edges which were kept after the reduction was sufficient for 3D reconstruction. On the other hand, revisiting the reduction step may be necessary for difficult image sets. This is in principle possible and is a part of our future work together with fine tuning of the priority keys assigned to different tasks.

Acknowledgements

This research was supported by the EC under Project HUMAVIPS FP7-ICT-247525 and by Czech Government under the research program MSM6840770038.

References

1. Schaffalitzky, F., Zisserman, A.: Multi-view matching for unordered image sets, or How Do I Organize My Holiday Snaps? In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 414–431. Springer, Heidelberg (2002)
2. Brown, M., Lowe, D.: Unsupervised 3D object recognition and reconstruction in unordered datasets. In: 3-D Digital Imaging and Modeling (3DIM), pp. 56–63 (2005)
3. Vergauwen, M., Van Gool, L.: Web-based 3D reconstruction service. Machine Vision and Applications (MVA) 17, 411–426 (2006)
4. Martinec, D., Pajdla, T.: Robust rotation and translation estimation in multiview reconstruction. In: CVPR 2007 (2007)
5. Snavely, N., Seitz, S., Szeliski, R.: Modeling the world from internet photo collections. IJCV 80, 189–210 (2008)
6. Snavely, N., Seitz, S., Szeliski, R.: Skeletal graphs for efficient structure from motion. In: CVPR 2008 (2008)

7. Agarwal, S., Snavely, N., Simon, I., Seitz, S., Szeliski, R.: Building Rome in a day. In: ICCV 2009, pp. 72–79 (2009)
8. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR 2006, vol. II, pp. 2161–2168 (2006)
9. Sivic, J., Zisserman, A.: Video Google: Efficient visual search of videos. In: Toward Category-Level Object Recognition (CLOR), pp. 127–144 (2006)
10. Li, X., Wu, C., Zach, C., Lazebnik, S., Frahm, J.: 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)
11. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. IJCV 42, 145–175 (2001)
12. Chum, O., Philbin, J., Isard, M., Zisserman, A.: Scalable near identical image and shot detection. In: Conference on Image and Video Retrieval (CIVR), pp. 549–556 (2007)
13. Guha, S., Khuller, S.: Approximation algorithms for connected dominating sets. Algorithmica 20, 374–387 (1998)
14. Havlena, M., Torii, A., Knopp, J., Pajdla, T.: Randomized structure from motion based on atomic 3D models from camera triplets. In: CVPR 2009, pp. 2874–2881 (2009)
15. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR 2007 (2007)
16. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). CVIU 110, 346–359 (2008)
17. Garey, M., Johnson, D.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman, New York (1979)
18. Nister, D.: A minimal solution to the generalized 3-point pose problem. In: CVPR 2004, pp. I: 560–567 (2004)
19. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
20. Schweighofer, G., Pinz, A.: Globally optimal O(n) solution to the PnP problem for general camera models. In: BMVC 2008 (2008)
21. Sturm, J.: SeDuMi: A software package to solve optimization problems (2006), http://sedumi.ie.lehigh.edu
22. Lourakis, M., Argyros, A.: The design and implementation of a generic sparse bundle adjustment software package based on the Levenberg-Marquardt algorithm. Tech. Report 340, Institute of Computer Science – FORTH (2004)
23. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. PAMI 13, 376–380 (1991)
24. Yahoo!: Flickr: Online photo management and photo sharing application (2005), http://www.flickr.com
25. Mičušík, B., Pajdla, T.: Structure from motion with wide circular field of view cameras. PAMI 28, 1135–1149 (2006)
26. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: BMVC 2002, pp. 384–393 (2002)
27. Irschara, A., Zach, C., Bischof, H.: Towards wiki-based dense city modeling. In: Virtual Representations and Modeling of Large-scale environments, VRML (2007)

Conjugate Gradient Bundle Adjustment

Martin Byröd and Kalle Åström*

Centre for Mathematical Sciences, Lund University, Lund, Sweden
{byrod,kalle}@maths.lth.se

Abstract. Bundle adjustment for multi-view reconstruction is traditionally done using the Levenberg-Marquardt algorithm with a direct linear solver, which is computationally very expensive. An alternative to this approach is to apply the conjugate gradients algorithm in the inner loop. This is appealing since the main computational step of the CG algorithm involves only a simple matrix-vector multiplication with the Jacobian. In this work we improve on the latest published approaches to bundle adjustment with conjugate gradients by making full use of the least squares nature of the problem. We employ an easy-to-compute QR factorization based block preconditioner and show how a certain property of the preconditioned system allows us to reduce the work per iteration to roughly half of the standard CG algorithm.

1 Introduction

Modern structure from motion (SfM) systems, which compute cameras and 3D structure from images, rely heavily on bundle adjustment. Bundle adjustment refers to the iterative refinement of camera and 3D point parameters based on minimization of the sum of squared reprojection errors and hence belong to the class of non-linear least squares problems. Bundle adjustment is important both as a final step to polish off a rough reconstruction obtained by other means as well as a way of avoiding accumulation of errors during an incremental reconstruction procedure.

A recent trend in SfM applications is to move from small and medium size setups to large scale problems (typically in the order 10^3-10^4 cameras or more), *cf.* [1,2,3,4]. Bundle adjustment in general has $\mathcal{O}(N^3)$ complexity, where N is the number of variables in the problem [5]. In the large scale-range of the spectrum, bundle adjustment hence starts to become a major computational bottleneck.

The standard algorithm for bundle adjustment is Levenberg-Marquardt with Cholesky factorization to solve the normal equations [6,7]. An interesting alternative to this is the method of conjugate gradients (CG), which has recently been applied in the context of bundle adjustment [8,1]. The conjugate gradient

* The research leading to these results has received funding from the swedish strategic research project Elliit, the European Research Council project Globalvision, the Swedish research council project Polynomial Equations and the Swedish Strategic Foundation projects ENGROSS and Wearable Visual Systems.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 114–127, 2010.

algorithm can be applied both as a non-linear optimization algorithm replacing Levenberg-Marquardt or as an iterative linear solver for the normal equations, where the latter approach seems to be the right choice for non-linear least squares.

In [8], which is of a speculative nature, the graph structure of the problem was used to derive multiscale preconditioners for bundle adjustment and conjugate gradients. While the authors show some promising preliminary results, they were not able to overcome some fundamental limitations yielding the preconditioners themselves expensive to construct and apply. In this paper we take a more straightforward approach and make use of the inherent sparsity structure of the problem to design a light-weight matrix based preconditioner. Doing bundle adjustment with conjugate gradients and block diagonal preconditioning was mentioned in the work of Agarwal *et al* on large scale structure from motion [1]. Compared to the work of Agarwal *et al*, where essentially the standard conjugate gradient algorithm was applied to the normal equations, the main novelty of this work is to make explicit use of the least squares nature of the problem for maximum efficiency and precision. Here we make use of the least squares property in several ways. Our main contributions are:

- We apply the CGLS algorithm (instead of the standard CG algorithm), which allows us to avoid forming $J^T J$, where J is the Jacobian, thus saving time and space and improving precision.
- A QR factorization based block-preconditioner, which can be computed in roughly the same time it takes to compute the Jacobian.
- We note that the preconditioned system has "property A" in the sense of Young [9], allowing us to cut the work per iteration in roughly half.
- An experimental study which sheads some new light on when iterative solvers for the normal equations may be successfully used.

2 Problem Formulation

We consider a setup with m cameras $C = (C_1, \ldots, C_m)$ observing n points $U = (U_1, \ldots, U_n)$ in 3D space. An index set $\mathcal{I}$ keeps track of which points are seen in which views by $(i, j) \in \mathcal{I}$ *iff* point j is seen in image i. If all points are visible in all views then there are mn projections. This is not the case in general and we denote the number of image points $n_r = |\mathcal{I}|$. The observation model $f(C_i, U_j)$ yields the 2D image coordinates of the point U_j projected into the view C_i. The input data is a set of observations $\hat{f}_{ij}$ such that

$$\hat{f}_{ij} = f(C_i, U_j) + \eta_{ij}, \tag{1}$$

where η_{ij} is measurement noise drawn from a suitable distribution. The unknown parameters $x = (C, U)$ are now estimated given the set of observations by adjusting them to produce a low re-projection error as realized in the following non-linear least squares problem

$$x^* = \underset{x}{\operatorname{argmin}} \sum_{(i,j)\in\mathcal{I}} \|\hat{f}_{ij} - f(C_i, U_j)\|^2. \tag{2}$$

The standard algorithm for dealing with non-linear least squares problem is the Gauss-Newton algorithm. Rewriting (2), our task is to solve the following optimization problem

$$x^* = \underset{x}{\operatorname{argmin}} \|r(x)\|^2, \tag{3}$$

where r is the vector of individual residuals $r_{ij} = f(C_i, U_j) - \hat{f}_{ij}$.

A first order expansion inside the norm in the non-linear sum of squares expression yields a linear least squares problem

$$\min_x \|r(x+\delta x)\|^2 \approx \|r(x) + J(x)\delta x\|^2, \tag{4}$$

where solving for δx in the usual least squares sense yields the equation for the update step:

$$J(x)^T J(x)\delta x = -J(x)^T r(x). \tag{5}$$

If the system matrix $J^T J$ does not have full rank, or if there are significant non-linearities then it is common to add a *damping* term λI to $J^T J$ and solve the damped system

$$(J^T J + \lambda I)\delta x = -J^T r. \tag{6}$$

We use the strategy by Nielsen [10] to update λ based on how well the decrease in error agrees with the decrease predicted by the linear model.

In the case of bundle adjustment, it is possible to partition the Jacobian into a camera part J_C and a point part J_P as $J = [J_C \; J_P]$, which gives

$$J^T J = \begin{bmatrix} J_C^T J_C & J_C^T J_P \\ J_P^T J_C & J_P^T J_P \end{bmatrix} = \begin{bmatrix} U & W \\ W^T & V \end{bmatrix}, \tag{7}$$

where U and V are block diagonal. One can now apply block wise Gaussian elimination producing the simplified system

$$(U - WV^{-1}W^T)\delta x_C = b_C - WV^{-1}b_P \tag{8}$$

and then substituting the obtained value of δx_C into

$$V\delta x_P = b_P - W^T \delta x_C \tag{9}$$

and solving for δx_P. This procedure is known as Schur complementation and reduces the computational load from solving a $(6m+3n)\times(6m+3n)$ system to solving a $6m \times 6m$ system followed by a quick substitution and block diagonal solve. In applications m is usually much smaller than n so this typically means substantial savings. For systems with up to a couple of hundred cameras, the most expensive step actually often lies in forming $WV^{-1}W^T$, since W is often quite dense. However, for larger problems the cost of solving the Schur system will dominate the computations. With the method of conjugate gradients we can avoid both Schur complementation and Cholesky factorization, thus avoiding the two dominant steps in terms if time and memory requirements. The price for this can be slower convergence especially near the optimum.

3 The Linear and Non-linear Conjugate Gradient Algorithms

The conjugate gradient algorithm is an iterative method for solving a symmetric positive definite system of linear equations

$$Ax = b, \tag{10}$$

introduced by Hestenes and Stiefel [11,12]. In its basic form it requires only multiplication of the matrix A with a vector, *i.e* no matrix-matrix multiplications and no matrix factorizations.

Conjugate Gradient Algorithm(x^0, A, b)

// An initial solution x^0 (possibly zero) has to be provided
$s^0 = b - Ax^0, p^0 = s^0, k = 0$
while $|s^k| >$ threshold
$\alpha^k = \frac{{s^k}^T s^k}{{p^k}^T A p^k}$
$x^{k+1} = x^k + \alpha^k p^k$
$s^{k+1} = s^k - \alpha^k A p^k$
$\beta^k = \frac{{s^{k+1}}^T s^{k+1}}{{s^k}^T s^k}$
$p^{k+1} = s^{k+1} + \beta^k p^k$
$k = k + 1$

The basic way to apply the conjugate gradient algorithm to the bundle adjustment problem is to form the normal equations $J^T J \delta x = -J^T r$ and set $A = J^T J, b = -J^T r$.

The linear CG method corresponds to minimization of the quadratic form $q(x) = \frac{1}{2} x^T A x - b^T x$. Fletcher and Reeves generalized the procedure to non-quadratic functions yielding the non-linear conjugate gradients algorithm [13]. Here, only the function $f(x)$ and its gradient $\nabla f(x)$ are available.

4 Conjugate Gradients for Least Squares

A naive implementation of the conjugate gradient algorithm for the normal equations would require forming $A = J^T J$ which is a relatively expensive operation. However, we can rewrite the updating formulas for α^k and s^{k+1} as

$$\alpha^k = \frac{{s^k}^T s^k}{(Jp^k)^T (Jp^k)}, \tag{11}$$

$$s^{k+1} = s^k - \alpha^k J^T (Jp^k), \tag{12}$$

implying that we only need to compute the two matrix-vector multiplications $w^k = Jp^k$ and $J^T w^k$ in each iteration. The resulting algorithm is known as CGLS [14]. The conjugate gradient method belongs to the wider family of Krylov

subspace optimizing algorithms. An alternative to CGLS is the LSQR algorithm by Paige and Saunders [15], which is based on Lanczos bidiagonalization. Mathematically CGLS and LSQR are equivalent, but LSQR has in some cases been observed to be slightly more stable numerically. However, in our bundle adjustment experiments these two algorithms have produced virtually identical results. Since LSQR requires somewhat more storage and computation than CGLS we have stuck with the latter.

5 Inexact Gauss-Newton Methods

As previously mentioned, there are two levels where we can apply conjugate gradients. Either we use linear conjugate gradients to solve the normal equations $J^T J dx = -J^T r$ and thus obtain the Gauss-Newton step or we apply non-linear conjugate gradients to directly solve the non-linear optimization problem.

Since $c(x) = r^T(x)r(x)$, we get $\nabla c(x) = -J^T(x)r(x)$ and we see that computing ∇c implies computing the Jacobian J of r. Once we have computed J (and r) we might as well run a few more iterations keeping these fixed. But, since the Gauss-Newton step is anyway an approximation to the true optimum, there is no need to solve the normal equations very exactly and it is likely to be a good idea to abort the linear conjugate gradient method early, going for an approximate solution. This leads to the topic of inexact Newton methods (see *e.g* [16] for more details). In these methods a sequence of stopping criteria are used to abort the inner iterative solver for the update step early. The logical termination quantity here is the relative magnitude of the residual of the normal equations $|s^k|$ (not to be confused with the residual of the least squares system r). A common choice is to terminate the inner CG iteration when

$$\frac{|s^k|}{|\nabla c(x_j)|} < \eta_j,$$

where the sequence $\eta_j \in (0, 1)$ is called a *forcing sequence*. There is a large body of research on how to select such a forcing sequence. We have however found the rule of thumb to select the constant $\eta_j = 0.1$ to provide a resonable trade off between convergence and number of CG iterations.

6 Preconditioning

The success of the conjugate gradient algorithm depends largely on the conditioning of the matrix A. Whenever the condition number $\kappa(A)$ is large convergence will be slow. In the case of least squares, $A = J^T J$ and thus $\kappa(A) = \kappa(J)^2$, so we will almost inevitably face a large condition number[1]. In these cases one

[1] Note that even if we avoid forming $A = J^T J$ explicitly, A is still implicitly the system matrix and hence it is the condition number $\kappa(A)$ we need to worry about.

can apply *preconditioning*, which in the case of the conjugate gradient method means pre-multiplying from left and right with a matrix E to form

$$E^T A E \hat{x} = E^T b,$$

where E is a non-singular matrix. The idea is to select E so that $\hat{A} = E^T A E$ has a smaller condition number than A. Finally, x can be computed from $\hat{x}$ with $x = E\hat{x}$. Often E is chosen so that EE^T approximates A^{-1} in some sense. Explicitly forming $\hat{A}$ is expensive and usually avoided by inserting $M = EE^T$ in the right places in the conjugate gradient method obtaining the *preconditioned conjugate gradient method*. Two useful preconditioners can be obtained by writing $A = L + L^T - D$, where D and L are the diagonal and lower triangular parts of A. Setting $M = D^{-1}$ is known as Jacobi preconditioning and $M = L^{-T} D L^{-1}$ yields Gauss-Seidel preconditioning.

6.1 Block QR Preconditioning

The Jacobi and Gauss-Seidel preconditioners alone do not make use of the special structure of the bundle adjustment Jacobian. Assume for a moment that we have the QR factorization of J, $J = QR$ and set $E = R^{-1}$. This yields the preconditioned normal equations

$$R^{-T} J^T J R^{-1} \delta\hat{x} = -R^{-T} J^T r,$$

which by inserting $J = QR$ reduce to

$$\delta\hat{x} = -R^{-T} J^T r$$

and $\delta\hat{x}$ is found in a single iteration step (δx is then be obtained by $\delta x = R^{-1}\delta\hat{x}$). Applying R^{-1} is done very quickly through back-substitution. The problem here is of course that computing $J = QR$ is exactly the sort of expensive operation we are seeking to avoid. However, we can do something which is similar in spirit. Consider again the partitioning $J = [J_C,\ J_P]$. Using this, we can do a block wise QR factorization in the following way:

$$J_C = Q_C R_C,\ J_P = Q_P R_P.$$

Due to the special block structure of J_C and J_P respectively we have

$$R_C = R(J_C) = \begin{bmatrix} R(\tilde{A}_1) & & & \\ & R(\tilde{A}_2) & & \\ & & \ddots & \\ & & & R(\tilde{A}_n) \end{bmatrix}$$

and

$$R_P = R(J_P) = \begin{bmatrix} R(B_1) & & & \\ & R(B_2) & & \\ & & \ddots & \\ & & & R(B_n) \end{bmatrix},$$

where

$$\tilde{A}_k = \begin{bmatrix} A_{k1} \\ A_{k2} \\ \vdots \\ A_{kn} \end{bmatrix}$$

and

$$B_k = \begin{bmatrix} B_{1k} \\ B_{2k} \\ \vdots \\ A_{mk} \end{bmatrix}$$

and where

$$A_{ij} = \partial_{C_i} r_{ij}, \;\; B_{ij} = \partial_{U_j} r_{ij}.$$

In other words, we can perform QR factorization independently on the block columns of J_C and J_P, making this operation very efficient (linear in the number of variables) and easy to parallelize. The preconditioner we propose to use thus becomes

$$E = \begin{bmatrix} R(J_C)^{-1} & \\ & R(J_P)^{-1} \end{bmatrix}.$$

Similar preconditioners were used by Golub *et al* in [17] in the context of satellite positioning. One can easily see that the QR preconditioner is in fact analytically equivalent to block-Jacobi preconditioning. Two important advantages are however that (i) we do not need to form $J^T J$ (as is the case with block-Jacobi) and (ii) that QR factorization of a matrix A is generally considered numerically superior to forming $A^T A$ followed by Cholesky factorization.

6.2 Property A

A further important aspect of the bundle adjustment Jacobian is that the preconditioned system matrix $\hat{J}^T \hat{J}$ has "property A" as defined by Young in [9].

Definition 1. *The matrix A has "property A"* iff *it can be written*

$$A = \begin{bmatrix} D_1 & F \\ F^T & D_2 \end{bmatrix}, \tag{13}$$

where D_1 and D_2 are diagonal.

The benefit is that for any matrix posessing "property A", the work that has to be carried out in the conjugate gradient method can roughly be cut in half as showed by Reid in [18]. This property can easily be seen to hold for $\hat{J}^T \hat{J}$:

$$\hat{J}^T \hat{J} = \begin{bmatrix} R(J_C) & \\ & R(J_P) \end{bmatrix}^{-T} \begin{bmatrix} J_C^T J_C & J_C^T J_P \\ J_P^T J_C & J_P^T J_P \end{bmatrix} \begin{bmatrix} R(J_C) & \\ & R(J_P) \end{bmatrix}^{-1} = \begin{bmatrix} Q_C^T Q_C & Q_C^T Q_P \\ Q_P^T Q_C & Q_P^T Q_P \end{bmatrix},$$

where $Q_C^T Q_C$ and $Q_P^T Q_P$ are both identity matrices and $Q_P^T Q_C = (Q_C^T Q_P)^T$. Partition the variables into camera and point variables and set $s^k = \begin{bmatrix} s_C^k \\ s_P^k \end{bmatrix}$. Applying Reid's results to our problem yields the following: By initializing so that $\delta x_C = 0$ and $\delta x_P = -J_P^T r$, we will have $s_C^{2m} = s_P^{2m+1} = 0$. We can make use of this fact in the following way (where for clarity, we have dropped the subscript j from the outer iteration):

INNER CG LOOP USING "PROPERTY A"(J, r)

$\eta = 0.1$
$\delta x_C^0 = 0,\ \delta x_P^0 = -J_P^T r,\ \hat{r}^0 = -r - J\delta x^0,\ p^0 = s^0 = J^T \hat{r}^0,$
$\gamma^0 = {s^0}^T s^0,\ q^0 = Jp^0,\ k = 0$
while $\|s^k\| > \eta \|s^0\|$

$$\alpha^k = \frac{\gamma^k}{{q^k}^T q^k}$$
$$\delta x^{k+1} = \delta x^k + \alpha^k p^k$$
$$\begin{cases} s_C^{k+1} = -\alpha^k J_C^T q^k,\ s_P^{k+1} = 0 \ k \text{ odd} \\ s_P^{k+1} = -\alpha^k J_P^T q^k,\ s_C^{k+1} = 0 \ k \text{ even} \end{cases}$$
$$\gamma^{k+1} = {s^{k+1}}^T s^{k+1}$$
$$\beta^k = \frac{\gamma^{k+1}}{\gamma^k}$$
$$p^{k+1} = s^{k+1} + \beta^k p^k$$
$$\begin{cases} q^{k+1} = \beta^k q^k + J_C s_C^{k+1} \ k \text{ odd} \\ q^{k+1} = \beta^k q^k + J_P s_P^{k+1} \ k \text{ even} \end{cases}$$

One further interesting aspect of matrices with "Property A" is that one can show that for these matrices, block-Jacobi preconditioning is always superior to Gauss-Seidel and SSOR preconditioners [14] (pages 286-287).

7 Experiments

For evaluation we compare three different algorithms on synthetic and real data. Standard bundle adjustment was performed using the Levenberg-Marquardt algorithm and sparse Cholesky factorization of the Schur complement to solve the normal equations. Cholesky factorization was performed using the Cholmod library with reverse Cuthill-McKee ordering. We henceforth denote this algorithm DBA for direct bundle adjustment. Secondly, we study a straight forward adaptation of the conjugate gradient algorithm to bundle adjustment by using $J^T J$ as the system matrix and the block diagonal of $J^T J$ as a preconditioner. We simply refer to this algorithm as CG. Finally, we denote the conjugate gradient method taylored to bundle adjustment as proposed in this paper CGBA for conjugate gradient bundle adjustment.

In all cases we apply adaptive damping to the normal equations as suggested in [10]. In the case of CGBA, we never form $J^T J$ and we instead apply damping by using the damped Jacobian

$$J_\lambda = \begin{bmatrix} J \\ \lambda I \end{bmatrix},$$

which can be factorized in the same manner as J for preconditioning.

For clarity, we focus on calibrated cameras only in this work. Including additional parameters such as focal length and distortion parameters presents no problem and fits into the same general framework without modification.

7.1 Synthetic Data: When Is the CG Algorithm a Good Choice?

A common statement is that standard bundle adjustment is good for small to medium size problems and that Conjugate Gradients should probably be the way to go for large and sparse problems. This is not quite true as we will show with a couple of synthetic experiments. In some cases CG based bundle adjustment can actually be a better choice for quite small problems. On the other hand it might suffer from hopelessly slow convergence on some large very sparse setups. Theoretically, the linear CG algorithm converges in a number of iterations proportional to roughly the square root of the condition number and a large condition number hence yields slow convergence. Empirically, this happens in particular for sparsely connected structures where unknowns in the camera-structure graph are far apart. Intuitively such setups are much less "stiff" and can undergo relatively large deformations with only little effect on the reprojection errors.

To capture this intuition we have simulated two qualitatively very different scenarios. In the first setup, points are randomly located inside a sphere of radius one centered at the origin. Cameras are positioned uniformly around the sphere at around two length units from the origin pointing roughly towards the origin. There are 10 times as many points as cameras and each camera sees 100 randomly selected points. Due to this, each camera shares features with a large percentage of the other cameras. In the second experiments, points are arranged along a circular wall with cameras on the inside of the wall pointing outwards. There are four points for each camera and due to the configuration of the cameras, each camera only shares features with a small number of other cameras. For each scenario we have generated a series of configurations with increasingly many cameras and points. One example from each problem type can be seen in Figure 1. For each problem instance we ran both standard bundle adjustment with Cholesky factorization and the Conjugate Gradient based bundle adjustment procedure proposed in this paper until complete convergence and recorded the total time. Both solvers produced the same final error up to machine precision. Since the focus of this experiment was on iterative versus direct solvers, we omitted the comparision CG method. The results of this experiment are perhaps somewhat surprising. For the sphere problem, CGBA is orders of magnitude faster for all but the smallest problems, where the time is roughly equal. In fact, the empirical time complexity is almost linear for CGBA whereas DBA displays the familiar cubic growth. For the circular wall scenario the situation is reversed. While CGBA here turns out to be painfully slow for the larger examples, DBA seems perfectly suited to the problem and requires not much more than linear time in the number of cameras. Note here that the Schur

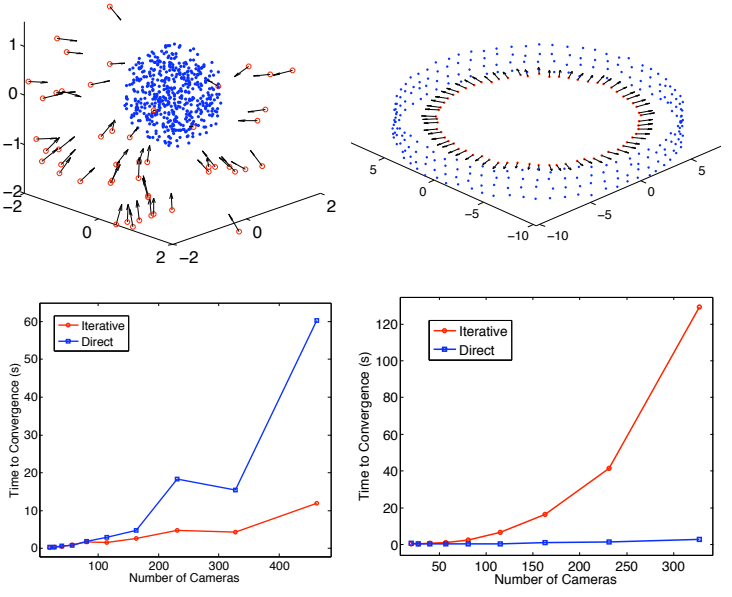


Fig. 1. Top-left: An instance of the sphere problem with 50 cameras and 500 3D points. Top-right: Points arranged along a circular wall, with 64 cameras viewing the wall from the inside. Bottom-left: Time to convergence vs. number of cameras for the sphere problem. This configuration is ideally suited to CG based bundle adjustment which displays approximately linear complexity. Bottom-right: Time vs. problem size for the circular wall. The CG based solver takes very long to converge, whereas the direct solver shows an almost linear increase in complexity, far from the theoretical $\mathcal{O}(N^3)$ worst case behaviour.

complement in the sphere setup is almost completely dense whereas in the wall case it is extremely sparse. The radically different results on these data sets can probably understood like this. Since the CG algorithm in essence is a first order method "with acceleration", information has to flow from variable to variable. In the sphere case, the distance between cameras in the camera graph is very small with lots of connections in the whole graph. This means that information gets propagated very quickly. In the wall problem though, cameras on opposite sides of the circular configuration are very far apart in the camera graph which yields a large number of CG iterations. For the direct approach "stiffness" of the graph does not matter much. Instead fill-in during Cholesky factorization is the dominant issue. In the wall problem, the Schur complement will have a narrow banded structure and is thus possible to factorize with minimal fill-in.

8 Community Photo Collections

In addition to the synthetic experiments, we have compared the algorithms on four real world data sets based on images of four different locations downloaded from the internet: The St. Peters church in Rome, Trafalgar square in London,

the old town of Dubrovnik and the San Marco square in Venice. The unoptimized models were produced using the systems described in [19,20,1].

The models produced by these systems initially contained a relatively large number of outliers, 3D points with extremely short baselines and very distant cameras with a small field of view. Each of these elements can have a very large impact on the convergence of bundle adjustment (both for iterative and direct solvers). To ensure an informative comparison, such sources of large residuals and ill conditioning were removed from the models. This meant that approximately 10% of the cameras, 3D points and reprojections were removed from the models.

In addition, we used the available calibration information to calibrate all cameras before bundle adjustment. In general this gave good results but for a very small subset of cameras ($< 0.1\%$) the calibration information was clearly incorrect and these cameras were removed as well from the models.

For each data set we ran bundle adjustment for 50 iterations and measured the total time and final RMS reprojection error in pixels. All experiments were done on a standard PC equipped with 32GB of RAM to be able to process large data sets. For the CG based solvers, we used a constant $\eta = 0.1$ forcing sequence and set the maxium number of linear iterations to 100. The results can be found in Table 1. Basically, we observed the same general pattern for all four data sets. Due to the light weight nature of the CG algorithms, these showed very fast convergence (measured in seconds) in the beginning. At a certain point close to the optimum however, convergence slowed down drastically and in none of the cases did either of the CG methods run to complete convergence. This is likely to correspond to the bound by the condition number of the Jacobian (which we were not able to compute due to the sizes of these problems). In other words, the CG algorithms have problems with the eigenmodes corresponding to the smallest singular values of the Jacobian. This situation makes it hard to give a fair comparison between direct BA and BA based on an iterative linear solver. The choice has to depend on the application and desired accuracy. In all cases,

Table 1. Performance statistics for the different algorithms on the four community photo data sets

Data set	m	n	n_r	Algorithm	Total Time	Final Error (Pixels)
St. Peter	263	129502	423432	DBA	113s	2.18148
				CGBA	441s	2.23135
				CG	629s	2.23073
Trafalgar Square	2897	298457	1330801	DBA	68m	1.726962
				CGBA	18m	1.73639
				CG	38m	1.75926
Dubrovnik	4564	1307827	8988557	DBA	307m	1.015706
				CGBA	130m	1.015808
				CG	236m	1.015812
Venice	13666	3977377	28078869	DBA	N/A	N/A
				CGBA	230m	1.05777
				CG	N/A	N/A

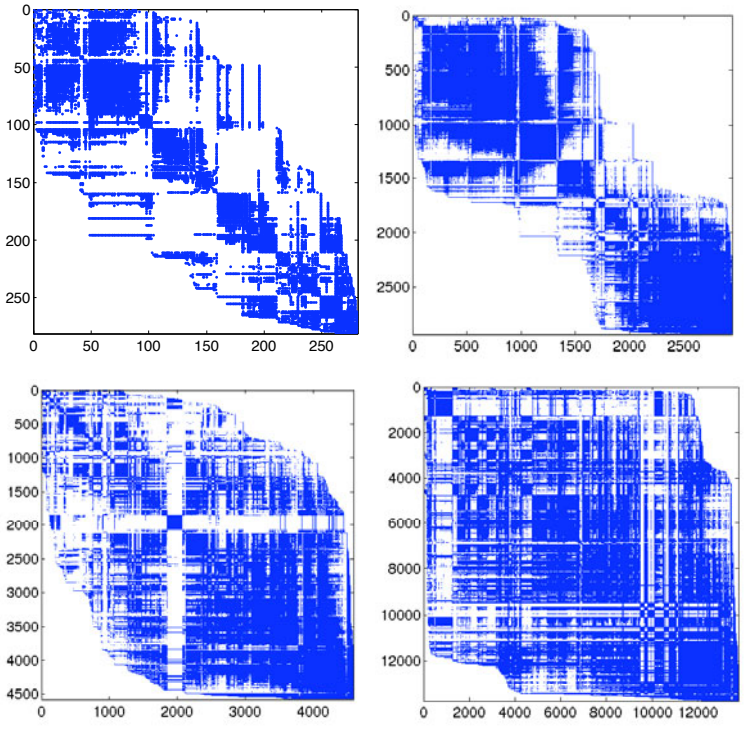


Fig. 2. Sparsity plots for the reverse Cuthill-McKee reordered Schur complements. Top-left: St. Peter, top-right: Trafalgar, bottom-left: Dubrovnik, bottom-right: Venice.

CGBA was about two times faster than CG as expected and in general produced slightly more accurate results.

For the Venice data set, we were not able to compute the Cholesky factorization of the Schur complement since we ran out of memory. Similarly, there was not enough memory in the case of CG to store both J and J^TJ. While Cholesky factorization in this case is not likely to be feasible even with considerably more memory, a more clever implementation would probably not require both J and J^TJ and could possibly allow CG to run on this instance as well. However, as can be seen from the other three examples, the relative performance of CG and CGBA is pretty constant so this missing piece of information should not be too serious.

As observed in the previous section, problem structure largely determines the convergence rate of the CG based solvers. In Figure 2, sparsity plots for the Schur complement in each of the four data sets is shown. To reveal the structure of the problem we applied reverse Cuthill-McKee reordering (this reordering was also applied before Cholesky factorization in DBA), which aims at minimizing the bandwidth of the matrix. As can be seen, this succeeds quite well in the case of St. Peter and Trafalgar. In particular in the Trafalgar case, two almost independent sets are discovered. As discussed in the previous section, this is

a disadvantage for the iterative solvers since information does not propagate as easily in these cases. In the case of Dubrovnik and in particular Venice, the graph is highly connected, which is beneficial for the CG solvers, but problematic for direct factorization.

9 Conclusions

In its current state, conjugate gradient based bundle adjustment (on most problems) is not in a state where it can compete with standard bundle adjustment when it comes to absolute accuracy. However, when good accuracy is enough, these solvers can provide a powerful alternative and sometimes the only alternative when the problem size makes Cholesky factorization infeasible. A typical application would be intermediate bundle adjustment during large scale incremental SfM reconstructions. We have presented a new conjugate gradient based bundle adjustment algorithm (CGBA) which by making use of "Property A" of the preconditioned system and by avoiding $J^T J$ is about twice as fast as "naive" bundle adjustment with conjugate gradients and more precise. An interesting path for future work would be to try and combine the largely orthogonal strengths of the direct versus iterative approaches. One such idea would be to solve a simplified (skeletal) system using a direct solver and use that as a preconditioner for the complete system.

References

1. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building rome in a day. In: Proc. 12th Int. Conf. on Computer Vision, Kyoto, Japan (2009)
2. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from Internet photo collections. Int. Journal of Computer Vision 80, 189–210 (2008)
3. Mordohai, A.F.: Towards urban 3d reconstruction from video (2006)
4. Cornelis, N., Leibe, B., Cornelis, K., Gool, L.V.: 3d urban scene modeling integrating recognition and reconstruction. Int. Journal of Computer Vision 78, 121–141 (2008)
5. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
6. Triggs, W., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment: A modern synthesis. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) ICCV-WS 1999. LNCS, vol. 1883, p. 298. Springer, Heidelberg (2000)
7. Lourakis, M.I.A., Argyros, A.A.: Sba: A software package for generic sparse bundle adjustment. ACM Trans. Math. Softw. 36, 1–30 (2009)
8. Byröd, M., Åström, K.: Bundle adjustment using conjugate gradients with multiscale preconditioning. In: Proc. British Machine Vision Conference, London, United Kingdom (2009)
9. Young, D.M.: Iterative solution of large linear systems. Academic Press, New York (1971)
10. Nielsen, H.B.: Damping parameter in marquardt's method. Technical Report IMM-REP-1999-05, Technical University of Denmark (1999)

11. Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. Journal of Research of the National Bureau of Standards 49, 409–436 (1952)
12. Golub, G.H., van Loan, C.F.: Matrix Computations, 3rd edn. The Johns Hopkins University Press, Baltimore (1996)
13. Fletcher, R., Reeves, C.M.: Function minimization by conjugate gradients. The Computer Journal 7, 149–154 (1964)
14. Björck, Å.: Numerical methods for least squares problems. SIAM, Philadelphia (1996)
15. Paige, C.C., Saunders, M.A.: Lsqr: An algorithm for sparse linear equations and sparse least squares. ACM Trans. Math. Softw. 8, 43–71 (1982)
16. Nocedal, J., Wright, S.J.: Numerical optimization, 2nd edn. Springer, Berlin (2006)
17. Golub, G.H., Manneback, P., Toint, P.L.: A comparison between some direct and iterative methods for certian large scale godetic least squares problems. SIAM J. Sci. Stat. Comput. 7, 799–816 (1986)
18. Reid, J.K.: The use of conjugate gradients for systems of linear equations possessing "property a". SIAM Journal on Numerical Analysis 9, 325–332 (1972)
19. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from internet photo collections. International Journal of Computer Vision 80, 189–210 (2007)
20. Snavely, N., Seitz, S.M., Szeliski, R.: Skeletal sets for efficient structure from motion. In: Proc. Conf. Computer Vision and Pattern Recognition, Anchorage, USA (2008)

NF-Features – No-Feature-Features for Representing Non-textured Regions

Ralf Dragon, Muhammad Shoaib, Bodo Rosenhahn, and Joern Ostermann

Institut fuer Informationsverarbeitung
Leibniz Universitaet Hannover
30167 Hannover, Germany
{dragon,shoaib,rosenhahn,ostermann}@tnt.uni-hannover.de

Abstract. In order to achieve a complete image description, we introduce no-feature-features (NF-features) representing object regions where regular interest point detectors do not detect features. As these regions are usually non-textured, stable re-localization in different images with conventional methods is not possible. Therefore, a technique is presented which re-localizes once-detected NF-features using correspondences of regular features. Furthermore, a distinctive NF descriptor for non-textured regions is derived which has invariance towards affine transformations and changes in illumination. For the matching of NF descriptors, an approach is introduced that is based on local image statistics.

NF-features can be used complementary to all kinds of regular feature detection and description approaches that focus on textured regions, i.e. points, blobs or contours. Using SIFT, MSER, Hessian-Affine or SURF as regular detectors, we demonstrate that our approach is not only suitable for the description of non-textured areas but that precision and recall of the NF-features is significantly superior to those of regular features. In experiments with high variation of the perspective or image perturbation, at unchanged precision we achieve NF recall rates which are better by more than a factor of two compared to recall rates of regular features.

1 Introduction

During the past years, the combination of interest point detector and local descriptor has been successfully applied in a high number of computer vision problems. The main reason for that is the fact that establishing local image

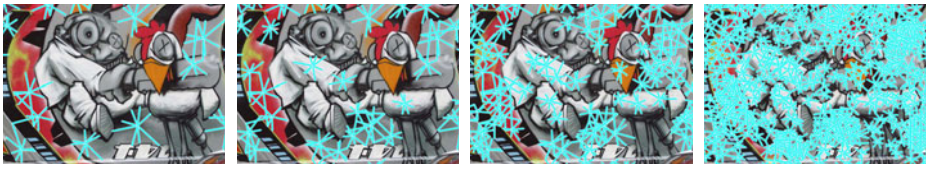


Fig. 1. Detected NF-features with increasing density using SIFT as regular features. The lines denote the extents that the feature descriptor is built from.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 128–141, 2010.

correspondences, which is one of the main computer vision problems, can be solved by inexpensive descriptor matching. As high repeatability under various external influences like changes in perspective or illumination is important for a stable matching, textured image regions with intensity variations in scale and space are chosen during the detection. Thereby, the local descriptor is created from high-entropy input data which results in distinctive descriptors.

However, broad categories of real-world objects have non-textured regions. Regular detectors very likely miss stable feature locations there. Additionally, most kinds of descriptors are built using local image gradients and thus lose their distinctiveness when built on non-textured areas. Hence, non-textured regions usually are not considered for detection and description.

In this paper we present a new feature type named *No-Feature features* (NF-features) that has the purpose to explicitly model regions without any features. This is inspired from the field of physics, in which the absence of electrons in conductors is modeled as the positively charged particle 'electron hole' with effective properties like mass and mobility. Likewise, NF-features are located on all those regions where any regular interest point detector left a 'feature hole'. Thus NF-features results in a complete description for all regions of an image which enhances detection and classification.

The **contributions** of this paper are

- the development of a detection method for NF-features ensuring there is a minimal density in the extracted features,
- the derivation of a new descriptor in which contrast and intensity-shift invariant image content is stored,
- the derivation of a statistical descriptor matching method which evaluates the local image noise variance, and
- comprehensive evaluation of NF in combination with SIFT, MSER & GLOH, Hessian-Affine & GLOH and SURF as regular features.
- For further evaluation of NF we provide binaries [1].

The remaining sections are structured as follows: In Section 2, we give an overview of related work and explain the differences to our approach. In Section 3, the algorithms used for NF detection, description and matching are described. In Section 4, we show experiments and give a conclusion in Section 5.

2 Related Work

For stability and repeatability, all commonly-used interest point detectors detect image content that contains high entropy. By evaluating the second moment matrix, the Harris Corner Detector [2] detects interest points with intensity gradients that vary in two directions and thus are precisely locatable. In [3] points with extremal intensities are detected and by the evaluation of surrounding image contents with rays, an affine orientation is assigned. Maximally Stable Extremal Regions (MSER) [4] are detected by finding connected components which are extremal as they either have lower or higher intensity values than all surrounding pixels. These regions can be considered homogeneous or non-textured, but in

order to be detected, they necessarily have to exhibit a significant contour. The following detectors which are based on the scale-space, detect blob-like features not only in the spatial but also in the scale domain. The Harris-Affine detector [5], which is based on the Harris Corner Detector, evaluates the second moment matrix at a given scale and thus locates anisotropic blobs in the image. The Hessian-Affine (HAff) [6] detector works similar but the Hessian matrix is evaluated instead. The famous SIFT (Scale Invariant Feature Transform) detector [7] uses the Difference of Gaussians operator to locate features that correspond to isotropic blobs in the unscaled image. In [8], generalized junction-type features were proposed as interest points which are detected at different scales. The idea of detecting all kinds of maxima that exhibit spatial unpredictability is exploited in [9], where regions with maximal salience are detected. For a stable localization in the scale-space, all these methods detect only significant maxima. As non-textured regions with non-elliptic shape result in blurred maxima in the scale-space, they are usually not considered as keypoint location.

In [10], the fusion of complementary information similar to our method has been proposed. They use a contour descriptor combined with a local descriptor and get improved results for the combination. However, non-textured regions are still not covered with this approach. As they contain no contours and no texture, no interest points may be described.

Local descriptors are usually built from statistical parameters around the detected location. The SIFT descriptor [7] is built from histograms of gradient orientation at the detected scale. [11] proposed GLOH (Gradient Location and Orientation Histogram) which extends SIFT by changing the location grid and using PCA for compression. [12] proposed SURF (Speeded Up Robust Features), an efficient variation of SIFT by using integral images for a more-efficient computation while making additional approximations towards SIFT. For invariance towards monotonic changes in intensity, SMD (Stable Monotonic Change Invariant Descriptor) [13] was introduced, which analyzes intensity order changes.

The idea of sampling lines for the descriptor that originate from the keypoint location is also used for Spiders [14] and for the intensity-based region detector [3]. In [14], the lines are used to determine the extents of a feature by evaluating the intensity run along a line and choosing that border location, where the intensity falls below a threshold for the first time. Likewise in [3], that location is selected where an intensity expression becomes maximal. Thus in contrast to NF, in both works the lines are used to determine the extents of a feature to make it affine invariant.

3 NF Features

Creating NF-features in an image I_1 follows the same paradigm as state-of-the-art local features: First the feature is detected and then a descriptor is built from the local image content. As our approach is always complementary to features like SIFT that we call *regular features*, the NF detection has to be performed after regular features have been detected. To match NF-features between images I_1

and I_2, we introduce a technique called second level matching, for which regular correspondences, NF features of I_1, and the image I_2 have to be given. Second level matching consists of second level detection and second level description which are both explained at the end of the following two sections.

In the derivation, x_c denotes the location of a feature of set F_c, features of images I_1 and I_2 are distinguished by $x_c^{(1)}$ and $x_c^{(2)}$, and $D(x, y)$ denotes the Euclidean distance between two vectors x and y.

3.1 NF Detection

NF-features should be complementary to regular features. Thus we detect an NF-feature at every location x_{nf} where all regular features at $x_{\mathrm{reg}} \in F_{\mathrm{reg}}$ are far, thus $D(x_{\mathrm{nf}}, x_{\mathrm{reg}}) > d_{\mathrm{far}}$. We define d_{far} as a constant factor c of the median nearest neighbor Euclidean distance d_{mnn} of the regular features:

$$d_{\mathrm{far}} = c \cdot d_{\mathrm{mnn}}(F_{\mathrm{reg}}) = c \cdot \mathrm{median}_{F_{\mathrm{reg}}} \left[D\left(x_{\mathrm{reg}}, \mathrm{nearest}_{F_{\mathrm{reg}}}(x_{\mathrm{reg}})\right)\right]. \quad (1)$$

Choosing $c = 3$ yields to a good trade-off between dense NF coverage and computation speed. Likewise, we clip d_{mnn} if it falls below 10 pel. In Fig. 1, detections with increasing NF density using d_{far} from $5d_{\mathrm{mnn}}$ down to $2d_{\mathrm{mnn}}$ are displayed.

The detection is performed iteratively using an algorithm similar to the Farthest Point Sampling [15]. Given F_{reg} and all already detected NF-features F_{nf}, we seek for the location which is farthest from all known features $F_{\mathrm{kf}} = F_{\mathrm{reg}} \cup F_{\mathrm{nf}}$. In other words, we seek for the center of the largest hole in F_{kf}. To find that location efficiently, the Delaunay triangulation is built for F_{kf} (cf. Fig. 2(a)). The edges of its dual graph, the Voronoi diagram, cover all points to which the distance to the nearest two neighbors is identical. Thus, Voronoi vertices cover all points which are locally farthest to all known feature locations. From all Voronoi vertices we choose that point with maximal distance d_{max} to F_{kf} as NF location. Features are located iteratively until d_{max} falls below d_{far}. Using this algorithm, we ensure no hole remains with a radius larger than d_{far}.

Keypoint detection should be robust and repeatable. However, this detection method is only repeatable if after a detection in image I_1, exactly the same regular features are detected in another image I_2. To overcome this, when regular correspondences $C_{\mathrm{reg}} = \{(F_{\mathrm{reg}}^{(1)}, F_{\mathrm{reg}}^{(2)})\}$ have been found between images I_1 and I_2, second level detection is performed. For each NF location $x_{\mathrm{nf}}^{(1)}$ from image I_1, a local transformation T to image I_2 is estimated using the nearest n regular features with correspondences in I_2 (cf. Fig. 2(b)). The local transformation $x^{(2)} = T(x^{(1)})$ is then applied to localize the NF-feature in image I_2 as

$$x_{\mathrm{nf}}^{(2)} = T\left(x_{\mathrm{nf}}^{(1)}\right). \quad (2)$$

Thus, the NF keypoint always fulfills the same local motion as the nearest regular correspondences. We call the involved regular features *anchor features*, as the NF keypoint is fixed to these features and performs the same local motion.

Assuming that the anchor features as well as the NF-feature are coplanar, T is a homography. For a stable estimate, RANSAC is used with the nearest

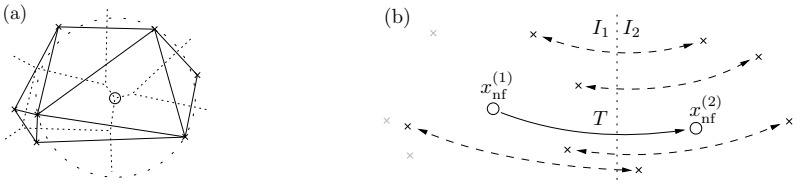


Fig. 2. (a) NF detection using the Delaunay triangulation (solid) and the Voronoi diagram (dotted). The NF-feature (small circle) is located at x_{nf}, which is that Voronoi vertex with maximum distance to all regular features (crosses). (b) Second level detection. Corresponding regular features from images I_1 and I_2 are used as anchor features in order to compute the local transformation $x^{(2)} = T(x^{(1)})$.

$n = 8$ correspondences, combined by a normalization step [16] and least-squares fitting. If RANSAC does not find a reasonable solution, there is no NF-feature located and thus no NF correspondence established. Likewise, any $x_{\mathrm{nf}}^{(2)}$ too close to a regular feature is not considered, as I_2 is assumed to be textured at that location. By this second level detection, we obtain candidates for the matching.

3.2 NF Description

The NF descriptor should specify the contents of the region around its location. As there was no regular feature detected in that region, it is very likely that it is non-textured. However, the hull of the area which is built from the nearest regular points is textured. We exploit this transition from dull to featured within the NF descriptor: The descriptor is created by analyzing intensity runs from the NF location to the nearest regular features. To describe the whole area around the NF location, it is divided into 8 segments of same angle (Fig. 3(a)). In each segment, a line is sampled which runs towards the nearest regular feature inside the segment. If no regular feature has been detected or it is very far, we sample along the segment middle with a distance of $5d_{\mathrm{mnn}}$, as we are unsure about the dimensions of the feature. By this sampling method, a hull around the NF location is formed with one hull point in each segment. We use the parametrization $t \in [0, 1]$ for the line starting at the NF location.

$$t_i = \frac{i}{N+1}, \quad i = 1 \ldots N\,. \tag{3}$$

To extract the deviation from a smooth transition from dull to featured, the difference between $I(t)$ and the linear transition $I_{\mathrm{lin}}(t) = I(0) + t \cdot (I(1) - I(0))$ is extracted for the descriptor (cf. Fig. 3(b)). For the jth segment, $j = 1 \ldots 8$, we receive N samples δ_{ij} along the line towards the hull point:

$$\begin{aligned} \delta_{ij} &= I(t_i) - I_{\mathrm{lin}}(t_i) && (4)\\ &= I(t_i) + (t_i - 1) \cdot I(0) - t_i \cdot I(1)\,. && (5) \end{aligned}$$

δ_{ij} is not contrast invariant, as $I^{(\mathrm{c})} = \alpha I$ yields $\delta^{(\mathrm{c})} = \alpha\delta$. As a stable measurement for normalization, we use the standard deviation of δ of all samples in the

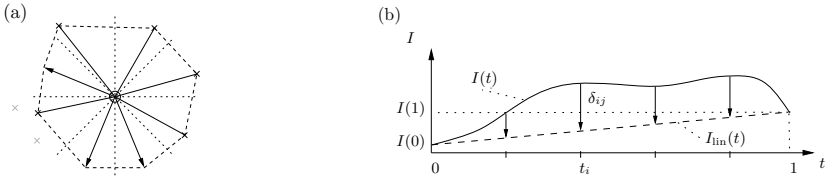


Fig. 3. (a) NF description by sampling lines (solid) from x_{nf} towards the hull points. The dotted lines denote the 8 segment borders. For every segment the nearest regular feature is selected as hull point. If no such feature exists, a point in the segment middle (arrows) with specific distance to x_{nf} is chosen as hull point. (b) Sampling δ_{ij} according to (5). The difference of the image intensity run $I(t)$ to the linear transition $I_{\mathrm{lin}}(t) = I(0) + t \cdot (I(1) - I(0))$ is extracted.

NF hull. For further processing, a zero-mean descriptor d_{ij} is required:

$$d_{ij} = \frac{\delta_{ij} - \operatorname{mean} \delta}{\sqrt{\operatorname{var} \delta}}\,. \tag{6}$$

The NF descriptor d_{ij}, which has variance 1 and mean 0, specifies the contents of the convex region spanned by the hull points. To store it in a memory-efficient way, each element is quantized. We evaluated that a uniform quantization of 8 bit with clipping at ± 2 does not change the descriptors significantly.

As the image texture is assumed to be non-textured near to the NF sampling locations, small localization errors during the sampling can be neglected. Besides, we can sample the low-passed image signal with acceptable loss in precision to reduce the noise variance by a factor of s_{lp}. We use a Gaussian-shaped filter of size 7×7 and a variance of 1 that performs a suppression of $s_{\mathrm{lp}} = 4.1$.

Computing the hull is not repeatable if other nearest regular features are used. Thus, we have to distinguish between first and second level descriptions again. When regular correspondences have been established and the NF-feature is located in image I_2, we also transform the hull points $x^{(1)}_{\mathrm{hull}}$ from I_1 to I_2 using $x^{(2)}_{\mathrm{hull}} = T(x^{(1)}_{\mathrm{hull}})$. Sampling is then performed analog to first level matching.

3.3 Descriptor Matching Using Local Noise Estimation

As the descriptor is invariant with respect to small offsets, the two possibilities for two descriptors to differ are that either the image content differs or the presence of image noise. The two cases are to be classified to consider two descriptors to differ or to match. Because the content of the NF hull is non-textured and thus very unlikely to contain any high-frequency patterns, it can be assumed that the local image variance is due to the noise. We use the high-pass filtered image signal around the NF location to estimate the variance of Gaussian-distributed image noise. For a more robust estimate, we collect estimations of the variance at all sampling locations and take the median value of all estimations.

To analyze if two descriptors match, the NF descriptor $d^{(1)}$ is compared element-wise with $d^{(2)}$ using the difference

$$e_{ij} = d_{ij}^{(1)} - d_{ij}^{(2)}. \tag{7}$$

If we assume that two segment descriptors do not match, $d^{(1)}$ and $d^{(2)}$ are two independent random variables which were each normalized in (6) to have $\operatorname{var} d = 1$. Assuming d_{ij} being Gaussian-distributed, e_{ij} is also Gaussian-distributed with $\operatorname{var} e = 2$. On the other side if we assume that the descriptors match perfectly, then $\operatorname{var} e = 0$. Among all perturbations which lead to non-perfect descriptor matches, image noise is the only one that is not due to image contents changes. Thus we further estimate the influence of image noise on e_{ij}.

First, we look at the influence of additive zero-mean Gaussian-distributed image noise with variance σ^2 on the samples $I(t)$ from (5). $I(0)$, $I(1)$ and $I(t_i)$ become independent random variables. During the sampling, their variance was reduced from the low-pass filter by s_{lp}. Thus, we have

$$\operatorname{var} I(t) = \frac{\sigma^2}{s_{\mathrm{lp}}}. \tag{8}$$

As the three random variables are scaled in (5) by factors of 1, $t_i - 1$ and t_i respectively, we get

$$\operatorname{var} \delta_{ij} = \frac{\sigma^2}{s_{\mathrm{lp}}} \left(1^2 + (t_i - 1)^2 + t_i^2\right) = 2\frac{\sigma^2}{s_{\mathrm{lp}}}(1 - t_i + t_i^2). \tag{9}$$

We assume that during the normalization of the descriptor in (6), the influence of the image noise on $\operatorname{var} \delta$, which was created from the whole descriptor, is negligible compared to the influence on δ_{ij}. The influence of σ^2 on d_{ij} is thus

$$\operatorname{var} d_{ij} = \frac{2\sigma^2}{s_{\mathrm{lp}} \operatorname{var} \delta}(1 - t_i + t_i^2) = 2\bar{\sigma}^2(1 - t_i + t_i^2), \tag{10}$$

where we introduce $\bar{\sigma}^2$ as normalized image noise variance. When we compare two independent descriptor elements according to (7), the variance of e_{ij} is

$$\operatorname{var} e_{ij} = 2(\bar{\sigma}_1^2 + \bar{\sigma}_2^2)(1 - t_i + t_i^2). \tag{11}$$

We can assume that the image noise variance is constant in the sampled area. Thus we can use (3) to derive the expected variance $E[\operatorname{var} e_{ij}]$ of the random variable e_{ij} over all realizations (i, j). An estimation of $E[\operatorname{var} e_{ij}]$ can be found by computing the element-wise mean square distance (MSD) between two zero-mean descriptors:

$$E[\operatorname{var} e_{ij}] = D_{\mathrm{MSD}}(d^{(1)}, d^{(2)}) = 2(\bar{\sigma}_1^2 + \bar{\sigma}_2^2)\frac{1}{N}\sum_{i=1}^{N}(1 - t_i + t_i^2) \tag{12}$$

$$= 2(\bar{\sigma}_1^2 + \bar{\sigma}_2^2) r_N, \tag{13}$$

where r_N is a scale factor ($\frac{3}{4} \leq r_N < \frac{5}{6}$) that increases with growing number of samples:

$$r_N = \frac{5N + 4}{6N + 6}. \tag{14}$$

With an a-priori probability of p for two matching line segments, we set the classification border to the weighted middle between the expected values.

$$b = 2(1-p) + 2p\left(\bar{\sigma}_1^2 + \bar{\sigma}_2^2\right) r_N \tag{15}$$

As there is no a-priori information about the area between the features, p is set to 0.5.

$$b = 1 + (\bar{\sigma}_1^2 + \bar{\sigma}_2^2) r_N \tag{16}$$

The theoretical limit where NF-features are not classifiable is at a mean image noise variance of

$$\sigma_{\max}^2 = \frac{s_{\mathrm{lp}} \operatorname{var} \delta}{2\, r_N}\,. \tag{17}$$

It seems we can handle all noisy images with a high noise suppression s_{lp} or with a high number of samples N, but the variance of δ will also decrease by this: As the image content of the sampled line is non-textured, the difference to the linear run sampled in (5) mainly contains low-frequency patterns. So the theoretical limit depends on the image contents and thus cannot be derived here.

We empirically determined $N = 4$ samples per line for regular camera images. Thus, we achieve the following descriptor size: when using the here-proposed parameters (8 segments with 4 samples per segment, 8 bit descriptor quantization) the descriptor only occupies 32 byte. With this approach, we have to additionally store the intra-segment angles (8 bit) and the distances of the 8 hull points (8 bit) as well as the normalized image variance with high precision (32 bit). Thus, the NF descriptor size is 52 byte.

4 Experiments

We first demonstrate the properties of NF in a cluttered environment (Fig. 4). Using NF-SIFT, we match 3 T-shirts worn by 6 different persons under different illumination conditions. It can be seen that regular features match only at the T-shirt logos whereas NF-features match on most of the T-shirt area. Further, occluded or changed image contents like the faces is not matched.

To show the performance of our algorithm, we use natural image pairs of sequences for the evaluation of affine invariant features from [11]. To demonstrate illumination invariance, we use the Memorial sequence from [17] originally used to create high dynamic range images. For a larger experiment, we use the Amsterdam Library of Object Images (ALOI) [18] which includes 1000 images under varying illumination conditions. To demonstrate NF-features are useful in combination with different kinds of feature types, we use SIFT, MSER, Hessian-Affine (HAff) and SURF as regular features, where for MSER and Hessian-Affine GLOH is used as descriptor.

As NF uses second level matching and as the NF descriptor comparison is not based on a nearest neighbor similarity, evaluating recall vs. precision graphs by varying thresholds is not suitable here. Instead we focus on the common way of establishing regular correspondences using second nearest neighbor (2NN)

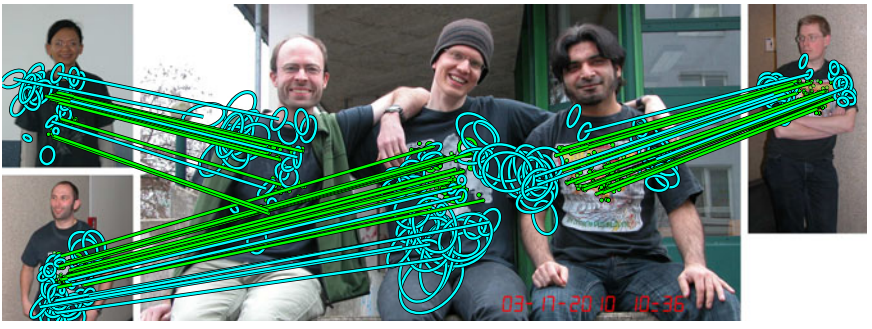


Fig. 4. NF (cyan) and SIFT (green) correspondences between 3 T-shirts worn by 6 different persons in a cluttered environment. The NF features are located on non-textured T-shirt regions. The correspondence lines are thinned out by a factor of 5.

matching as proposed in [7]. We then build NF correspondences, estimate them separately from the regular correspondences and compare all eight cases.

To measure performance, we use precision and recall of the extracted keypoints of the first image I_1 of every image pair (I_1, I_2) according to (18), where true positives (tp), false positives (fp) and false negatives (tp) are counted.

$$\text{recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}, \quad \text{precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}. \tag{18}$$

To verify the correspondences in the case of sequences with a moving camera, homography matrices supplied with the test material are used as ground truth. As we focus on dense object description for object recognition, we want to count an imprecise localization as inlier in contrast to 'real' outliers with false correspondences. For true positive correspondences we thus accept a maximum deviation of 15 pel from the ground truth which is approximately the average d_{mnn} of the ALOI image database. Correspondences with a higher distance are classified as false positives. All features detected only in the first image of the illumination image pair are counted as false negatives. True negatives are not analyzed as no significant occlusions exist in the sequences.

The runtime for processing NF-features using our non-optimized code depends on the image contents. If there are many regular features near each other (e.g. Fig. 6(b)) and large areas are unsampled, the iterative sampling algorithm from Section 3.1 covers large unsampled regions. In such cases, the runtime is up to 10 times the processing time of regular SIFT features. In images, where holes between regular features are filled (e.g. Fig. 1) NF matching needs roughly twice the processing time.

4.1 Descriptor Invariance in Image Sequences

First we examine the influence of changes in global illumination ('Leuven'), in changes of internal ('Bark', 'Boat') and external camera parameters ('Graf',

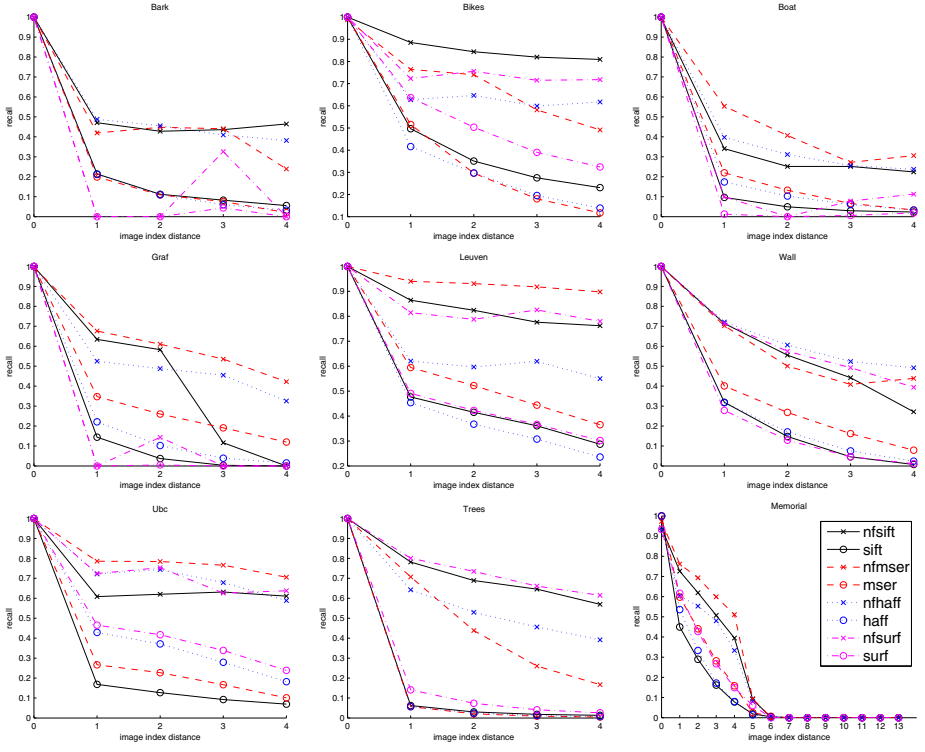


Fig. 5. Recall over the image index distance in the sequences 'Bark' (zoom and rotation), 'Bikes' (blur), 'Boat' (zoom and rotation), 'Graf' (viewpoint), 'Leuven' (illumination), 'Wall' (viewpoint), 'Ubc' (JPEG compression), 'Trees' (blur) and 'Memorial' (dynamic range). Crosses denote NF features, circles regular features.

'Wall'), in adding blur ('Bikes', 'Trees') and JPEG artifacts ('Ubc') and in variations of the dynamic range ('Memorial'). For each series, correspondences between all image pairs are established using 2NN for regular features and the here-presented methods for NF. To analyze the descriptor invariance, we reduce the effect of wrong second level detection due to false regular correspondences (We further analyze this in Section 4.3). We enforce high precision by a loose outlier filtering of the regular correspondences using RANSAC to estimate a homography from the unfiltered correspondences. By this the precision values are similar (almost all above 0.9). So we can compare the approaches using the recall value, which we average between all image pairs of the same image index distance (Fig. 5).

It can be seen that for all sequences NF always get higher recall values than regular features, often with a factor of more than two. This is positively surprising as regular features serve as anchor points for the second level detection. Thus we can deduce that NF descriptors have better invariance properties towards illumination changes, blurring, JPEG compression and affine transformations than regular features.

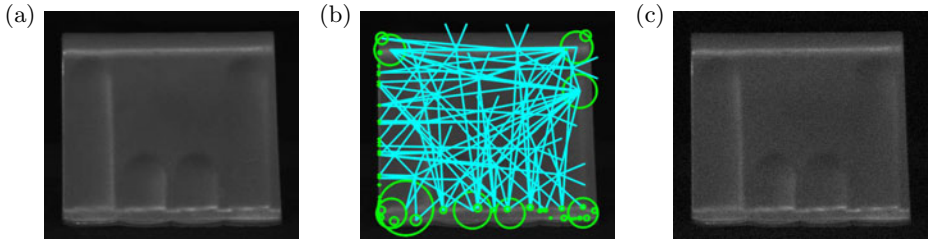


Fig. 6. (a) Original image 256 of the ALOI database. (b) The sampled lines of the detected NF-features (cyan) and SIFT anchor points (green circles denoting the extent). (c) The examined image with added image noise variance of 100.

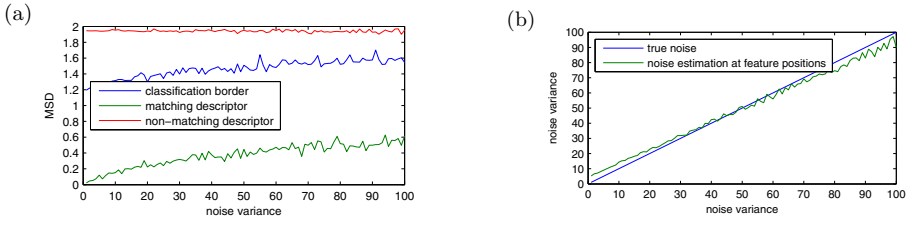


Fig. 7. (a) Influence of image noise on the descriptor difference variance from image 256 of the ALOI database. (b) Noise variance estimation at the feature locations vs. true noise variance.

4.2 Influence of Image Noise onto the Descriptor Distinctiveness

In this experiment, the distinctiveness of the NF descriptor is analyzed. Topological information from regular correspondences is not considered. In contrast to all kinds of regular state-of-the-art descriptors, NF descriptors are directly built from image intensity values. Thus the influence of image noise on the distinctiveness of the descriptor seems to be crucial. We now analyze the behavior of NF towards noise. Therefore we use the center view of object 256 of the ALOI database which shows a non-textured surface that has regular features at the borders only (Fig. 6). This means there is no transition from non-textured to textured during the line sampling for the descriptor. This is highly-crucial as the NF descriptor has low variance (cf. (17)) and by this image noise has a high impact on the matching result. We detect NF locations on the object and compute the descriptor differences while adding Gaussian image noise. Using the descriptor MSD, we compare descriptors which should match and those which should not match (Fig. 7(a)).

It can be seen that the descriptor difference variance runs as expected: For non-matching features it is independent from the image noise and reliably at approximately 2, where it grows from 0 with increasing noise if the features match. However, we have small systematic deviations from the derived model concerning the estimation of the image noise (cf. Fig. 7(b)). Besides, the MSD of non-matching features is slightly but significantly smaller than 2 which means

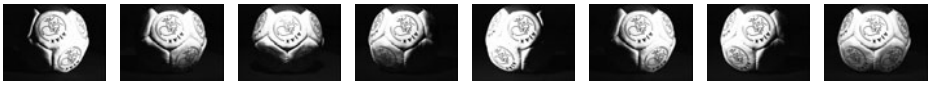


Fig. 8. ALOI object 103 with varying illumination $l_1 \dots l_8$ viewed from camera $c1$

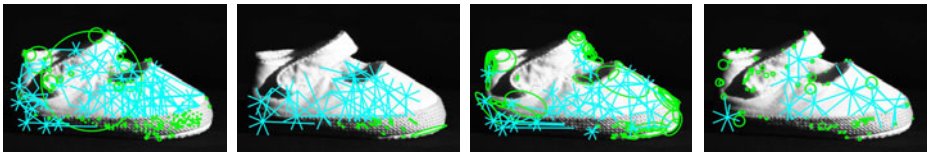


Fig. 9. Matching regular (green ellipses) and NF-features (cyan stars) of object 113 from the ALOI image database using NF-SIFT, NF-MSER, NF-HAff and NF-SURF

that they are statistically dependent. However these deviations are small and likely to be overestimated in this experiment because of the large untextured area. So we can deduce that NF descriptors are distinctive under the influence of image noise, even for small changes in image contents.

4.3 Image Database

We use the series 'Illumination Direction Collection' from ALOI with camera $c1$ in which one object is observed by a static camera during 8 different illumination conditions $l_1 \dots l_8$ (Fig. 8). Illumination $l_1 \dots l_5$ were taken at angles of $-60° \dots 60°$ in steps of $30°$. l_6 and l_7 were taken combining the side illuminations $l_1 + l_2$ and $l_4 + l_5$ respectively. l_8 is all illuminations combined.

We establish correspondences of each object illumination setting with each other illumination setting of the same object. To measure the impact of false NF correspondences due to false anchor point correspondences, we do not filter any correspondence like in Section 4.1. However, we allow the correspondence cluster filtering which is performed in SIFT, as it is an essential part of the algorithm. We compute recall and precision values according to (18) for all illumination pairs of the same object and average them. The results are plotted as precision and recall matrices over the eight illumination setting in Fig. 10 and in the form of precision and recall graphs over the angle of illumination change in Fig. 11.[1] In Fig. 9 we show a comparison for the detection of all examined NF feature combinations.

Generally two tendencies can be observed: Concerning the precision, images with similar illumination, e.g. (l_1, l_6), have higher precision values for regular features. With increasing variations of the illumination, NF outperforms regular features in precision, e.g. (l_1, l_5). However, NF-HAff show inferior results

[1] Please note that NF recall of identical images does not necessarily have to equal to 1, as the estimation of the local motion model T may fail if there are too few suitable anchor points available.

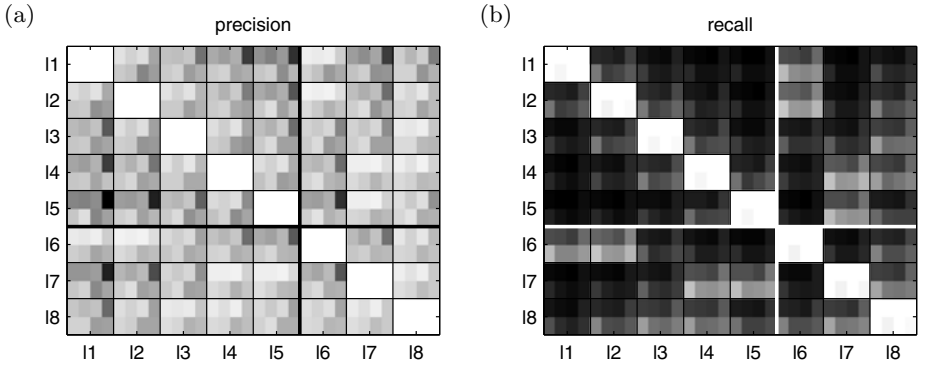


Fig. 10. Precision (a) and recall (b) matrix for the illumination conditions $l_1 \dots l_8$, where $l_1 \dots l_5$ are single-illuminated images and l_6, l_7 and l_8 are illuminated with a combinations thereof. In each square, regular (top) and NF features (bottom) are opposed using (starting left) SIFT, MSER, HAff and SURF. White denotes precision and recall of 1, black a precision of 0.72 and a recall of 0.06.

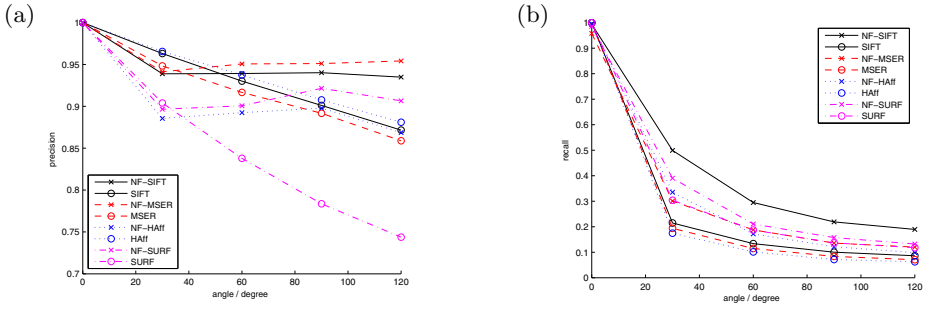


Fig. 11. Mean precision (a) and recall (b) over the angle of illumination change using NF-SIFT, SIFT, NF-MSER, MSER, NF-HAff, HAff, NF-SURF and SURF

compared to HAff. Concerning the recall rates, NF always outperforms regular features, where NF-SIFT is by far better than the other three NF combinations.

5 Conclusion

We derived a framework for NF-features which is complementary to every regular interest point detection approach with local descriptors. During detection, centers of regions unsampled by a regular feature detection are determined as NF locations. The second level matching algorithm re-locates suitable NF features in further images according to a local transformation which is extracted from already-established regular correspondences. The descriptor is built by sampling lines from the non-textured NF location to the nearest regular feature locations.

Using standard test material and enforcing high precision, we demonstrated that the repeatability of NF-features is significantly improved towards regular features, often by a factor of more than two. In a challenging experiment with high variations of the illumination without outlier filtering, we also achieved significantly better results concerning recall and precision. Thus, NF-features are not only useful for a complete description of the image contents but also improve recall and precision rates. For further evaluation, we provide binaries [1] that may be combined with any type of regular features.

References

1. NF Project Website, http://www.tnt.uni-hannover.de/project/nf
2. Harris, C., Stephen, M.: A combined corner and edge detector. In: Fourth Alvey Vision Conference, pp. 147–151 (1988)
3. Tuytelaars, T., van Gool, L.: Matching widely separated views based on affine invariant regions. IJCV 59(1), 61–85 (2004)
4. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable external regions. Image and Vision Computing (2004)
5. Mikolajcyk, K., Schmid, C.: An affine invariant interest point detector. In: ICCV (2002)
6. Mikolajcyk, K., Schmid, C.: Scale and affine invariant interest point detector. IJCV 60, 63–86 (2004)
7. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
8. Förstner, W., Dickscheid, T., Schindler, F.: Detecting interpretable and accurate scale-invariant keypoints. In: ICCV, pp. 2256–2263 (2009)
9. Kadir, T., Zisserman, A., Brady, M.: An affine invariant salient region detector. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 228–241. Springer, Heidelberg (2004)
10. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of adjacent contour segments for object detection. TPAMI 30(1), 36–51 (2008)
11. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. TPAMI 27(10), 1615–1630 (2005)
12. Bay, H., Tuytelaars, T., van Gool, L.: Surf: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
13. Gupta, R., Mittal, A.: Smd: A locally stable monotonic change invariant feature descriptor. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 265–277. Springer, Heidelberg (2008)
14. Stanski, A., Hellwich, O.: Spiders as robust point descriptors. In: Kropatsch, W.G., Sablatnig, R., Hanbury, A. (eds.) DAGM 2005. LNCS, vol. 3663, pp. 262–268. Springer, Heidelberg (2005)
15. Eldar, Y., Lindenbaum, M., Porat, M., Zeevi, Y.Y.: The farthest point strategy for progressive image sampling. TIP 6(9), 1305–1315 (1997)
16. Hartley, R.I.: In defense of the eight-point algorithm. TPAMI 19, 580–593 (1997)
17. Debevec, P.E., Malik, J.: Recovering high dynamic range radiance maps from photographs. In: SIGGRAPH, pp. 369–378 (1997)
18. Geusebroek, J.M., Burghouts, G.J., Smeulders, A.W.M.: The Amsterdam Library of Object Images. IJCV 61(1), 103–112 (2005)

Detecting Large Repetitive Structures with Salient Boundaries

Changchang Wu[1], Jan-Michael Frahm[1], and Marc Pollefeys[2]

[1] Department of Computer Science
UNC Chapel Hill, NC, USA
{ccwu,jmf}@cs.unc.edu
[2] Department of Computer Science
ETH Zürich, Switzerland
marc.pollefeys@inf.ethz.ch

Abstract. This paper presents a novel robust and efficient framework to analyze large repetitive structures in urban scenes. A particular contribution of the proposed approach is that it finds the salient boundaries of the repeating elements even when the repetition exists along only one direction. A perspective image is rectified based on vanishing points computed jointly from edges and repeated features detected in the original image by maximizing its overall symmetry. Then a feature-based method is used to extract hypotheses of repetition and symmetry from the rectified image, and initial repetition regions are obtained from the supporting features of each repetition interval. To maximize the local symmetry of each element, their boundaries along the repetition direction are determined from the repetition of local symmetry axes. For any image patch, we define its repetition quality for each repetition interval conditionally with a suppression of integer multiples of repetition intervals. We determine the boundary along the non-repeating direction by finding strong decreases of the repetition quality. Experiments demonstrate the robustness and repeatability of our repetition detection.

1 Introduction

Repetition and symmetry are frequently used in the design of urban architecture. In fact, buildings often consist of a hierarchy of repetitions and symmetries (e.g. Fig. 1). Particularly, most of the basic repeating elements on facades (such as doors and windows) are symmetric by themselves, repetition and symmetry coexist and interplay at different scales. This paper introduces a new method to detect repeating elements with salient boundaries in facade images.

The symmetry and repetition patterns together with the appearance of the repeating/symmetric elements provide a strong characterization of the scene. Given that, particularly for urban scenes, the symmetries and repetitions of a scene describe its high-level structure, they can be used for wide baseline matching. One area where this representation would be useful is in the reconstruction from urban photo collections as in [1]. The reliable boundaries of the detected repeating elements and the symmetric structure can be used as compact image

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 142–155, 2010.

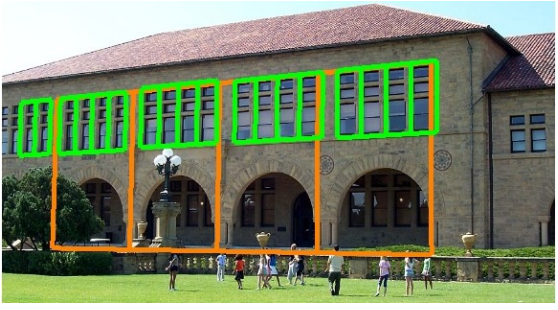


Fig. 1. Example of our detected repetitive structures. Note that the vertical boundaries are selected automatically to distinguish between the interesting elements and high frequency repetition of the roof.

features for effective recognition. Since such structures encode significantly more scene semantics than, for example, SIFT features [2], the matching is significantly less ambiguous. The known scene symmetries and repetitions allow us to automatically extract the facade grammars [3,4,5] as well as the semantic parsing of the images. Additionally, the known structure of the facades allows to regenerate facades based its grammar or compensate for occlusions by replacing occluded parts through their symmetric or repetitive equivalent.

Reliably detecting repetitions and extracting their boundaries is a significantly challenging problem. Even though images of planar facades can be rectified to a frontal view by using the vanishing points of the facade, the appearance of repeating elements may still significantly change, due to reflections and occlusions. In addition, the perspective change for non-planar structures on the facade plane severely affects the local symmetries.

A particularly challenging scenario that draws our attention is where the large repetitive structures repeat only along the horizontal direction (e.g. Fig. 1). Homogeneous regions, edges along vanishing directions, and high-frequency repetitions cause additional ambiguities in choosing meaningful boundaries for the repeating elements. To reliably detect the boundaries of such structures, we need to distinguish between regions that belong to different repetition groups (with different repetition intervals).

The remainder of the paper is organized as follows. Section 2 briefly discusses the related work. Section 3 discusses the few of our observations on repetition in urban scenes. Section 4 gives our vanishing point detection and sparse repetition analysis. Sections 5 introduces our repetition quality functions. Section 6 proposes our dense repetition detection algorithm with salient boundary detection. Experiments are discussed in Section 7 and conclusions are given in Section 8.

2 Related Work

Repetitions are usually hypothesized from the matching of local image features, and repetition are often detected as a set of sparse repeated features by growing

or tracking from the small sets of initial features towards their immediate spatial neighbors [6,7,8,9,10,11]. Dense detection of repetition requires the determination of the boundaries of repeating elements. Liu et. al [12] determine the boundary of repeating elements by maximizing the local symmetries. A limitation of their method is the requirement of a 2D repetition grid, which are not always available in urban environment. This paper shares their idea of maximizing local symmetries, but beyond that we separate different repetition groups by evaluating the local repetition quality conditionally for different repetition intervals.

Additional assumptions about the shape of the repeating elements are sometimes used to define the boundaries of repeating elements. Korah et. al [13] assume the repeating elements to be rectangular and extract them based on the edge segments in the rectified images. Their assumptions is often not completely valid in urban scenes because curved structures are very common. Our method uses a less restrictive assumptions only requiring the repeating elements to be approximately symmetric.

The general symmetry includes translational symmetry (we refer to as repetition), reflective symmetry and rotational symmetry. Many researches have proposed frameworks that can solve both translational symmetry and reflective symmetry(e.g. [14], [9]). Our method also handles both but in a joint fashion. We use the coexistence of repetition and symmetry to define the boundaries for our detected repetition regions along the repeating direction.

Perhaps most closely related to this paper is the work of Müller et. al [15]. They also aim to recover the architectural grammar describing the structure of the facade. The results are impressive, but require significantly stronger assumptions than for our approach. Besides rectification as in our approach, this approach requires a tight rectangular boundary delineating the facade (which seems to be a manual step as no automated solution is provided). It is further assumed that within this region vertical repetitions occur over the whole width and horizontal repetitions over the whole height. This is more restrictive than the bottom up approach we propose in this paper which only requires local support. As [15] only demonstrates their approach in the presence of both horizontal and vertical repetitions this seems to be required. Our approach works in the presence of horizontal repetition (or symmetry) alone. Finally, in [15] boundaries between elements are chosen based on edge support and distance heuristics and can yield undesired results. An important contribution of our work is to propose a principled approach to determine those boundaries based on the symmetry assumption and on direct image support. Beyond the scope of our paper, [15] refines the subdivision of facade elements and enables manual depth adjustments to yield detailed 3D facade reconstructions which is ideally suited for rendering.

3 Observations and Assumptions

Urban scenes are often designed with many repetitive structures, this section lists some observations that guided the design of our detection algorithm.

1. Dominant repetition(s) are mostly along the vanishing point direction(s) with equal 3D spacing. This gives us the opportunity to refine the vanishing point(s) based on repetition;
2. While many existing approaches require 2D repetition, many buildings lack vertical repetitions and symmetries. Our approach is specifically developed to handle this case.
3. Repeating architectural elements typically also exhibit reflective symmetry around vertical axes. Symmetry axes occur at twice the frequency of the repetition, in the middle and in between repeated elements. We use this to localize the vertical boundary between repeating elements (up to a two-fold ambiguity). Only in very few cases have we observed buildings where this principle is not satisfied. Note that the rectangle structure assumptions used for example by [13,15] is a special case of this assumption.

4 Sparse Repetition and Symmetry Detection

In this paper, we denote the extraction of repetition and symmetry from keypoints as sparse detection. This section first introduces our improved vanishing point detection, and then discusses our sparse detection in the rectified images.

4.1 Vanishing Point Refinement by Maximizing Overall Symmetry

Accurate vanishing point (VP) detection is important in our framework because we assume the repetitions along vanishing directions. Inaccuracy in VP locations will disturb the finding of optimal repetition interval and symmetry axes since the pairwise distances between the matched features change gradually. In our approach we use the cascaded hough transform [16] to compute the vertical and one or more horizontal vanishing points from edge pixels as initialization.

We propose a VP refinement by maximizing the overall symmetry in the entire image using both edges and features. Given a pair of horizontal and vertical vanishing points, VP_H and VP_V, a homography $T = T(VP_H, VP_V)$ can be determined to rectify the image. We define the transformation to keep the original resolution as much as possible to avoid too much shrinking and expanding. By matching SIFT [2] features extracted in the original image along both vanishing directions and keeping the closest matches (closest in the image), three sets of feature pairs can be extracted. We use R_H for horizontal repetition, R_S for horizontal symmetry, R_V for vertical repetition.

Consider a set of point pairs $R \in \{R_H, R_V, R_S\}$ in the original image and a transformation T, we use $X^T(R)$ to denote the distribution of their horizontal distances after rectification, $Y^T(R)$ for the distribution of their rectified vertical distances and $C^T(R)$ for the distribution of the horizontal coordinates of their rectified midpoints. Typically in urban scenes images, there exist only a few strong symmetry axes and repetitions intervals. Correspondingly, we expect to see only a few strong peaks in the histogram of $X^T(R_H)$, $Y^T(R_V)$ and $C^T(R_S)$. These strong peaks correspond to the minimum information that are required to represent most of the data distribution. Therefore, we expect low entropies

from those histograms. This paper optimizes the rectification by minimizing the summed entropy, so that the vanishing directions are better aligned with repetition directions and symmetry axes.

We use H to denote the entropy function. It can be proven that $H(X^T(R_H))$ and $H(Y^T(R_V))$ are invariant to any affine transformations, and $H(C^T(R_S))$ is invariant to transformation in the form of $\begin{pmatrix} a & 0 \\ b & c \end{pmatrix}$. However, such an affine ambiguity can be resolved by using the point distances in the direction that is perpendicular to repetition or symmetry, $Y^T(R_H)$, $X^T(R_V)$ and $Y^T(R_S)$, because they are only invariant to transformations in the form of $\begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix}$ given a finite resolution of histogram.

Consider a distance distribution $D(x) \in \{Y^T(R_H), X^T(R_V), Y^T(R_S)\}$ that are expected to be close to zero, we use the entropy of $D(x) + D(-x)$ in our minimization to reduce both drift from zero mean and large variance. We denote such entropy function by $\widehat{H}$. In our case, this can apply to $Y^T(R_H)$, $Y^T(R_S)$, and $X^T(R_V)$. Optionally, the edge information can be incorporated straightforwardly. Given the set of edge segments G_H and G_V that are corresponding to the two vanishing point, $Y^T(G_H)$ and $X^T(G_V)$ can be used the same as repetition.

By assuming the different distributions independent of each other and ignoring their joint distributions, we define an energy function for the repetition and symmetry of an image as

$$\begin{aligned} Q(VP_H, VP_V) = {} & H(X^T(R_H)) + H(Y^T(R_V)) + H(C^T(R_S)) + \widehat{H}(Y^T(R_H)) \\ & + \widehat{H}(Y^T(R_S)) + \widehat{H}(X^T(R_V)) + \widehat{H}(Y^T(G_H)) + \widehat{H}(X^T(G_V)) \end{aligned}$$

and the vanishing points VP_H, VP_V are then recovered at the minimum as

$$(VP_H, VP_V) = \underset{VP_H, VP_V}{\operatorname{argmin}} Q(VP_H, VP_V) \tag{1}$$

It can be seen that our method still optimizes both vanishing points when vertical repetition R_V is missing because the horizontal symmetry constraints the vertical vanishing points. Liu [17] has pointed out the potential of using symmetry in rectification, which were used by [7] to rectified facade images of 2D repetition grids. Our paper goes beyond to work with more general cases.

In this paper, individual entropies are weighted by the number of points to avoid bias from small point sets, and gradient descent is used to solve Eq 1. Our experiments show the VP refinement significantly reduces the drift of the estimated repetition interval when the initial detection is not accurate enough.

4.2 Repetition Intervals and Symmetry Axes

With the detected VPs, the original images are rectified to be fronto-parallel, and afterwards upright SIFT features are extracted (similar to the concept of U-SURF [18]). The single fixed orientation for all features is a natural choice given that the rotation is compensated through the rectification. Hence, our

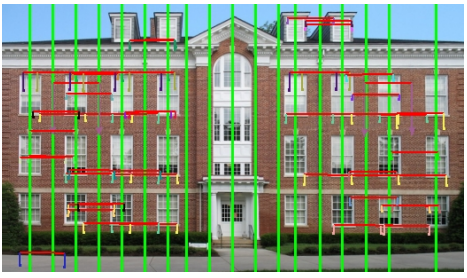


Fig. 2. An example of detected repeating features and symmetry axes. Only the feature pairs for the strongest repetition interval are displayed. It can be seen that the symmetry axes are repeating at half the interval of the window repetition.

feature matching does not suffer under descriptor changes from the erroneous orientation detections. The upright SIFT features are then matched along the horizontal and vertical direction. Note that the feature matching for reflective symmetry detection uses the mirrored matching [8].

By matching features along the horizontal direction and vertical direction, histograms of possible horizontal repetition intervals, vertical repetition intervals, and symmetries can be obtained from the features pairs. Local maxima are extracted from histograms to get a set of repetition intervals $\{I\}$ and symmetry axes $\{\Lambda\}$. In this paper, we do not try to recover vertical symmetries since they typically do not show up in urban scenes. We also skip any repetition intervals that are smaller than 30 pixels to focus on only large repetitive structures.

For each repetition interval and symmetry axis, the bounding box of their matches features gives rough regions for the repetition and symmetry. Unfortunately these regions are often inaccurate due to noise in their appearance and the ambiguity caused by small repetitive structures. To find the correct region, a dense measurement should be used.

Consistent with our assumption #3, the local symmetries and the symmetries between neighbouring repeating elements repeat with an interval of half of the structure size. See Fig. 2 for an example. Selecting the horizontal boundaries at the position of those symmetry axes maximizes the local symmetry of the repeating elements.

5 Evaluation of Repetition Quality

In order to define salient boundaries for repeating elements, we need to densely evaluate how well each location fits the repetition interval under consideration. While it is important to have some invariance to lighting changes and other small variations, non-repeating elements have to be identified. In addition, it is also important to suppress spurious support that could come from homogeneous regions and repetitions at higher frequencies (for example, the roof eaves in Fig. 2 has a repetition interval of $\frac{1}{5}$ of the window distances). We first use

image patches to evaluate the similarity between any two locations. In order to be invariant to scale changes and different rectification, the patch size W_I is selected proportionally to the repetition interval I. Through our experiments we have determined that $W_I = \frac{I}{4}$ consistently provides good results.

To provide robustness to small variations and lighting, patch similarity is evaluated by comparing SIFT descriptors, which is effeciently computed on GPU [19]. Given a repetition interval I and a location x, we use $D_R(x, I)$ to denote the distance between the normalized SIFT descriptor at x and $x + I$. Similarly, the matching distance wrt. a symmetry axis Λ is denoted as $D_S(x, \Lambda)$.

It can be verified that if an element is repeated many times, then if I is a valid repetition interval $2I, 3I, \ldots$ will also be valid. Therefore, we are interested in the smallest valid repetition interval and want to suppress its multiples. It is therefore important to verify that for a repetition interval I, the repetition intervals $\{\frac{I}{2}, \frac{I}{3}, \ldots\}$ are not valid repetition intervals. In fact, this would only have to be verified for $\frac{I}{p}$ with p prime numbers. In practice, verifying for the first few prime numbers is sufficient (we go up to 7). Notice that this automatically also covers the issue of homogeneous regions as those would verify repetition for any interval. Inspired by the widely used ratio test in SIFT matching, we choose a set of translations $T_I = \{0, \pm\frac{I}{2}, \pm\frac{I}{3}, \ldots\}$, compute the set of matching distances for them $V = \{D(x, I + t)|t \in T_I\}$, and define the following quality function

$$f(x, I) = \min(\alpha \frac{V_{(2)} + \sigma}{D(x, I) + \sigma}, 1) \tag{2}$$

where $V_{(2)}$is the second smallest distance in V. α is a parameter used for truncating the quality so that the quality function evaluates to 1 when $D(x, I)$ is significantly smaller than $V_{(2)}$ (we use $\alpha = 0.7$ as typical for the SIFT ratio test [2]). Adding a small number σ reduces noise when all distances are very small, which can be seen as a variance in the SIFT distance distribution (we use $\sigma = 0.1$). It can be seen that $f(x, I) > \alpha$ only when I is local minimum. Note that the definition works for both single patch or a patch set.

In feature matching, a small ratio between the smallest distance and the second often corresponds to a high probability of being a correct match [2], and such a ratio test filters out both ambiguous matches and poor matches. Similarly, a high $f(x, I)$ indicates high probability of x being salient repetition for interval I. The quality measure will be low if appearances change too much or if a patch matches better under other intervals or everywhere. This strategy gives penalty to both noise and ambiguous high frequency regions (e.g. Fig. 3).

As evaluation of single patches is very noisy, we define similarity and quality measures to evaluate repetition for image regions. The distance between two patch sets is defined as its the median distance: $D_R(X, I) = \text{median}\{D_R(x, I)|x \in X\}$. For quality function, we use a pre-learned threshold[1] T to select a inlier patch set $X_I\{x|x \in X, D_R(x, I) < T\}$ of an image region X, and use the inlier set to evaluate the repetition quality as $F(X, I) = f(X_I, I)$. Our experiments show that this quality function is very robust to outliers even for low inlier ratios.

[1] $T = 0.64$ learned from the distributions in labeled images is used in this paper.

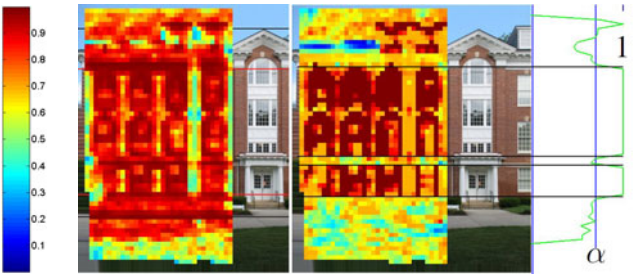


Fig. 3. Our similarity and quality measurement. The colored-patches in the left image gives the distance map (The visualization uses $1-\frac{1}{2}d^2$ to map distance $[0, \sqrt{2}]$ to $[0, 1]$). The colored patches in the right image gives the quality map and the curve gives the quality for each row. The distance map shows good matching for the grass, roof eaves and the horizontal edges, but our quality function is able to penalize them. The black lines in the right image gives the places where the vertical boundary are detected.

In order to avoids unreliable evaluation from noise, we set the quality measure to 0 when inlier ratio is less than 20%.

To correctly handle the first and last element of a repetition sequence, we define the bidirectional distance and quality

$$D^+(X, I) = \min(D(X, I), D(X, -I))$$
$$f^+(X, I) = \max(f(X, I), f(X, -I))$$

The distance map and quality map refer to D^+ and f^+ unless specified otherwise. Similar with $F(X, I)$, only inliers $X_I^+ = \{x | x \in X, D^+(x, I) < T\}$ are considered while evaluating f^+ for a patch set instead of a single patch.

6 Dense Detection

Our dense detection uses the detected sparse repetition and symmetry to obtain their initial regions, and refines them by dense matching and propagation.

6.1 Region Initialization and Propagation

It is a natural choice to select the horizontal boundaries of the repeating elements according the detected repeating symmetry axes (e.g. Fig. 2) since such boundaries generate elements with maximal local symmetry. As illustrated in Fig. 4, the initial horizontal extent of repetition region is defined by a group of symmetry axes that have horizontal distances of $\frac{I}{2}$ or I with each other. The initial vertical range is chosen to cover the matched feature pairs whose line segments intersect with the symmetry axes.

Detecting repeating elements is more complicated than detecting symmetry because the larger repetition count requires propagation and verification in order to get the full correct regions. Due to perspective change and noise, not all

Algorithm 1. The Repetition Detection Algorithm

```
Detect vanishing points and rectify image.
Find sparse repetitions {I} and symmetry axes {Λ}  (Section 4)
for each un-processed repetition interval I do
    Find sets of repeating symmetry axes {Λ_I}
    while {Λ_I} is not empty do
        Find a consecutive set of axes with gap I/2 or I
        Initialize region from the symmetry axes (Section 6.1)
        Propagate the region by matching at interval I
        Find region boundaries and sub-regions. (Section 6.2)
        Search and analyze vertical repetition.
        Find further decompositions of regions.  (Section 6.3)
        Save detected repeating elements
        Remove covered symmetry axes from {Λ_I}
    end while
    Mark repetitions that can be modeled as processed
end for
```

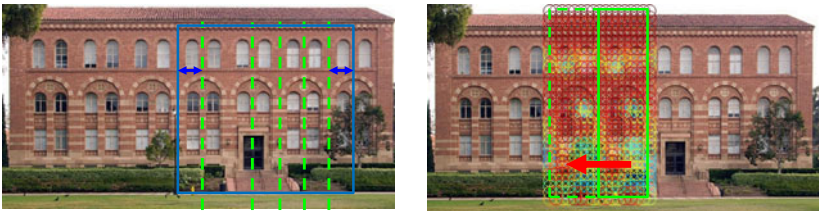


Fig. 4. Our region initialization from symmetry and propagation by dense matching

symmetry axes can be perfectly detected from initial feature matching. The initialization in previous step is likely to miss some parts of the actual repetition region. To extend the repetition region horizontally, we take steps of $\pm I$ or $\pm\frac{I}{2}$ to match a rectangular region of width I at the desired location. If the inlier ratio for both the left and right $\frac{I}{2}$ are high enough, the region is extended by the step size. Given the large window sizes, it is actually not necessary to match all the pixels. Typically, a sparse grid of locations can be used instead like Fig. 4.

6.2 Boundary Detection

Without using vertical repetition, we select the vertical boundaries based on the quality evaluation of row scanlines. Basically, we exclude regions that lack salient repetitions at interval I by simply setting boundaries where the quality of rows $F^+(X, I)$ drops from 1 to α (e.g. roof eaves and grass in Fig. 3).

With the determined vertical boundaries, multiple repeating elements can be defined after filtering out the rows without salient repetition between them.

After the horizontal repetition analysis, sparse vertical repetition analysis is applied in the detected region, and the boundaries for the vertical repetition are

Fig. 5. Example of decomposition. The color stripes in the left image shows the continuation score, and the black vertical lines give the detected element boundary edges. The right image shows the final decomposition as 4 different repetition groups.

then detected from vertical repetition quality map in a similar way. The initial region is then decomposed to sub-regions that have both horizontal and vertical repetition, and sub-regions that have only horizontal repetition.

6.3 Decomposition

As shown in Fig. 5, the possible mistakes of initializing from symmetry axes is the over-grouping of different repeating elements that have the same repetition intervals. In this case, the matching distances between neighboring elements will change over the entire horizontal range, it particularly gets large matching distances at the places where the repetition elements change. We define a continuation score function to evaluate how the repetition continues over a range of 4 times the repetition interval. Similar with the quality function, we define a continuation score from X to $X + I$ based on the ratio of distances

$$Cont(X, I) = \frac{\min(D(X_I^+ - I, I), D(X_I^+ + I, I)) + \sigma}{D(X_I^+, I) + \sigma}$$

The ratio threshold α used in repetition quality functions is basically a closeness threshold. For regions where we have good continuation of repetition, the continuation score should be in $(\alpha, 1/\alpha)$. At places where the repetition changes to something else, there will be much smaller continuation score. We particularly look for local minimums along horizontal direction that satisfy

$$Cont(X, I) < \min(Cont(X - I, I), Cont(X + I, I), \alpha)$$

Such local minimums give the possible locations that separate different repetition elements, and connecting such points vertically defines the edges between different repetition elements. To be robust to noises, we use regions of size $I \times I$ to evaluate the continuation score. Fig. 5 gives an example of the continuation score and the resulting decomposition.

7 Experiments

This section presents our qualitative results and quantitative results. We run our experiments with the same setting on all the results included in this paper.

7.1 Qualitative Results

Fig. 6 gives a few of our detection results. It can be seen that our detection algorithm robustly finds salient boundaries for both horizontal direction and vertical direction. The boundary detection is robust to occlusions, illumination changes, perspective changes, and existence of homogeneous regions and high-frequency repetition regions. As shown in of 2, 3, 4, 9, 13, 18 of Fig. 6 our algorithm detects vertical boundaries based on our quality function and correctly generates multiple repetition regions vertically.

Although our algorithm initializes the regions from symmetry axes, we do not enforce strong symmetry constraint on the detected elements. This allows the repetition detection under very large viewpoints, where the symmetry is

Fig. 6. Detection shown in the original images. Best viewed in color with 4× zoom.

Table 1. Our detection performance on the ZübuD dataset

Category	#	Percentage
No detection due to VP detection failure	25	4%
No detection due to other algorithmic limitations	34	5%
Partial Detection; Missing major repetitions	88	12%
Full detection of all major repetitions; Some boundaries errors	67	9%
Full detection of all major repetitions; Good boundaries	509	70%

very weak (e.g. 4, 9, 14, 16 in Fig. 6). In such cases, the repeating elements are detected with imperfect symmetries, and the horizontal boundaries may not be optimal. The experiments also show several of our limitations. In Fig. 6.3, the repetition from the left tower to the right tower is missing because the repetition interval is much larger than the tower width. Our current proportional patch size will not work, unless the ratio is allowed to vary. Fig. 6.8 does not detect the pure vertical repetition on the right side because our implementation currently only looks for vertical repetitions for horizontally repeating elements. We do have small errors in boundary detection like in Fig. 6.4 where too much is occluded for correct boundary detection. Fig. 6.17 has detected a wrong repetition due to inaccuracy of the second vanishing point pair.

7.2 Quantitative Evaluation

We use the ZübuD database [20] to evaluate out detection. ZübuD contains 1005 images of 201 buildings in Zürich taken from different viewpoints and illuminations conditions. We first manually filtered out 282 images that do not have clear repetitions that satisfy our assumptions (Due to occlusions, curved surface, etc). Fig. 6.17-19 are 4 examples from ZübuD. Table 7.2 is the statistics of our detection on the 723 remaining images. It can be seen that our algorithm has high successes rate for both VP detection and repetition detection.

Furthermore, we run an image retrieval experiment to evaluate the repeatability of our detection. We select the 140 buildings that have clear repetitions on at least 4 images. Our algorithm detects 10096 features in total (eacch element is counted as one; average 14 per images). Similar with SIFT descriptor, for each repeating element, we compute a 4x4 and a 8x8 gradient orientation histogram grid aligned with repeated elements to get a 128D resp. 512D feature descriptor. Paticularly, uniform weighting is used instead of Gaussian weighting to give equal importance to each cell. The distance of a feature to an image is defined as its smallest distance to all the features in that image. Given a single feature, images can be retrieved by selecting the closest ones. In this experiment, a feature-image retrieval is considered correct if the image is one of the other 4 images of the same building. For comparison, we select the 10/100 SIFT features that have the largest scales in each image to run the same experiment. Fig. 7 shows the retrieval precisions for the first 4 nearest neighbors, where our detection of repeating elements demonstrates relatively high repeatability. It is worth

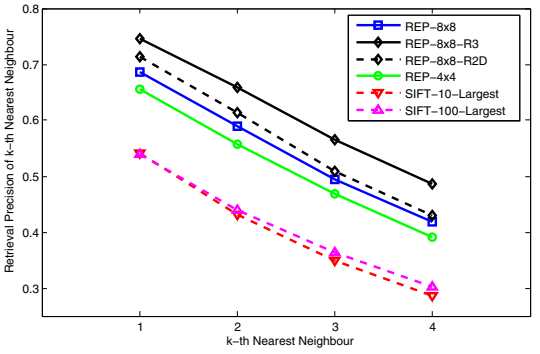


Fig. 7. Evaluation by single-feature image retrieval. REP refers to our repeating elements. 8x8 and 4x4 refers to the grid size for feature descriptor. R3 refers to the elements that repeat at least 3 times. R2D refers to the features that belong some 2D repetition grids. It can be seen that our repetition-based features achive better repeatability compared with standard image features. We believe that further improvements can be achieved with a new descriptor to capture more details. Additionally, features in R2D and R3 have better precision because they are easier to detect.

pointing out that many of the retrieval failures are due to the similar structures (especially windows) on different buildings.

8 Conclusion and Future Work

We propose a novel method to detect repeating elements on architectural facades. The main contributions are the new boundary selection for the dense repetition detection. We initialize our detection from symmetry axes to maximize the local symmetry. We also propose a quality function to conditionally evaluate how image patches fit a repetition interval, which leads to accurate vertical boundary detection. Our method is very efficient by evaluating repetition and symmetry with adaptiveness to the scale of repetitions. Typical images require only 2-4 seconds to complete the full analysis with the help of GPU. We evaluate our detection on large datasets and demonstrate the robustness and repeatability of our algorithm. Our method works particularly well for low-count and purely horizontal repetitions which has not been addressed by most previous work.

In future work, we hope to use the proposed repetition and symmetry detection scheme to automatically extract architectural grammars from images. We also hope to be able to recover missing 3D information by finding gradual changes of repetition and symmetry at different depths and generate true ortho-photos of facades from oblique views. Due to perspective changes, repeating elements at different depth that have a same 3D repetition interval will show different 2D repetition intervals in a rectified image (e.g. 9 and 14 in Fig. 6). Building further on the preliminary experiment presented in the evaluation, an interesting area of future work is to use the repetition/symmetry regions as invariant feature extractor and develop specific appearance and repetition descriptors.

References

1. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3d. ACM Transactions on Graphics (SIGGRAPH Proceedings) (2006)
2. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110 (2004)
3. Ripperda, N., Brenner, C.: Application of a formal grammar to facade reconstruction in semiautomatic and automatic environments. In: Proc. of the 12th AGILE Conference on GIScience (2009)
4. Wonka, P., Wimmer, M., Sillion, F., Ribarsky, W.: Instant architecture. In: SIGGRAPH 2003 (2003)
5. Mueller, P., Wonka, P., Haegler, S., Ulmer, A., Gool, L.V.: Procedural modeling of buildings. In: SIGGRAPH 2006 (2006)
6. Leung, T., Malik, J.: Detecting, localizing and grouping repeated scene elements from an image. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1064, pp. 546–555. Springer, Heidelberg (1996)
7. Schindler, G., Krishnamurthy, P., Lublinerman, R., Liu, Y., Dellaert, F.: Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In: CVPR. IEEE Computer Society Press, Los Alamitos (2008)
8. Loy, G., Eklundh, J.O.: Detecting symmetry and symmetric constellations of features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 213–225. Springer, Heidelberg (2006)
9. Wenzel, S., Drauschke, M., Förstner, W.: Detection of repeated structures in facade images. Pattern Recognition and Image Analysis 18, 406–411 (2008)
10. Schaffalitzky, F., Zisserman, A.: Geometric grouping of repeated elements within images. In: BMVC 1999 (1999)
11. Park, M., Collins, R., Liu, Y.: Deformed lattice discovery via efficient mean-shift belief propagation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 474–485. Springer, Heidelberg (2008)
12. Liu, Y., Collins, R.T., Tsin, Y.: A computational model for periodic pattern perception based on frieze and wallpaper groups. IEEE Transactions on PAMI 26, 354–371 (2004)
13. Korah, T., Rasmussen, C.: 2D lattice extraction from structured environments. In: ICIP 2007, vol. II, pp. 61–64 (2007)
14. Turina, A., Tuytelaars, T., Van Gool, L.: Efficient grouping under perspective skew. In: CVPR 2001, vol.1, pp. I-247–I-254 (2001)
15. Mueller, P., Zeng, G., Wonka, P., Gool, L.V.: Image-based procedural modeling of facades. In: Proceedings of ACM SIGGRAPH 2007/ACM Transactions on Graphics, vol. 26. ACM Press, New York (2007)
16. Tuytelaars, T., Van Gool, L., Proesmans, M., Moons, T.: The cascaded hough transform as an aid in aerial image interpretation. In: Proc. ICCV, pp. 67–72 (1998)
17. Liu, Y., Collins, R.: Skewed symmetry groups. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 872–879 (2001)
18. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
19. Wu, C.: SiftGPU: A GPU implementation of scale invariant feature transform (SIFT) (2007), http://cs.unc.edu/~ccwu/siftgpu
20. Shao, H., Svoboda, T., Gool, L.V.: ZuBuD — Zürich buildings database for image based recognition. Technical Report 260, Computer Vision Laboratory, Swiss Federal Institute of Technology (2003)

Fast Covariance Computation and Dimensionality Reduction for Sub-window Features in Images

Vivek Kwatra and Mei Han

Google Research, Mountain View, CA 94043

Abstract. This paper presents algorithms for efficiently computing the covariance matrix for features that form sub-windows in a large multi-dimensional image. For example, several image processing applications, *e.g.* texture analysis/synthesis, image retrieval, and compression, operate upon *patches* within an image. These patches are usually projected onto a low-dimensional feature space using dimensionality reduction techniques such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), which in-turn requires computation of the covariance matrix from a set of features. Covariance computation is usually the bottleneck during PCA or LDA ($O(nd^2)$ where n is the number of pixels in the image and d is the dimensionality of the vector). Our approach reduces the complexity of covariance computation by exploiting the redundancy between feature vectors corresponding to overlapping patches. Specifically, we show that the covariance between two feature components can be reduced to a function of the relative displacement between those components in patch space. One can then employ a lookup table to store covariance values by relative displacement. By operating in the frequency domain, this lookup table can be computed in $O(n \log n)$ time. We allow the patches to *sub-sample* the image, which is useful for hierarchical processing and also enables working with filtered responses over these patches, such as local *gist* features. We also propose a method for fast projection of sub-window patches onto the low-dimensional space.

1 Introduction

We consider the problem of efficiently computing the covariance matrix for feature vectors that can be expressed as sub-windows in a large image. This problem occurs in construction of codebooks for image patches, where each patch (sub-window) in the image is projected to a low-dimensional space using a dimensionality reduction technique such as Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA). This low-dimensional representation is then useful for several tasks such as matching (search for patches with matching feature vectors in texture analysis/synthesis, example-based super-resolution, non-local image denoising and inpainting), compression (using Vector Quantization), and detection/recognition (e.g. face recognition using wavelet features).

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 156–169, 2010.

Sub-window features may not be limited to 2D images, but also useful in 1D time-series such as audio signals and for 3D analysis in volumetric data or video.

We present an algorithm for efficiently computing the covariance matrix from these sub-window features by exploiting the redundancy between overlapping windows. Specifically, we show that the covariance between two feature components can be expressed as a function of the relative displacement between those components in patch space. This further reduces to a cross-correlation operation which can be computed quickly in frequency domain. Using a similar analysis, the projection of sub-window features onto the low-dimensional PCA or LDA basis can also be expressed as a cross-correlation (or filtering) operation, and therefore computed efficiently.

We are particularly motivated by texture analysis and synthesis tasks, where image patches or their filtered representations are used as descriptors of local image texture. Recent work on scene analysis employs *gist* descriptors for images [21]. The *local* version which computes gist features for sub-images and provides textural information for similar patch search is also based on sub-window features. Computing these descriptors requires learning weights for filter bank responses of the image. An intermediate step involves performing PCA over features representing sub-windows in the filtered response images. Due to the high dimensionality of these feature vectors, image windows are usually sub-sampled before performing PCA. However, using our approach, PCA can be performed efficiently without resorting to sub-sampling.

In example-based synthesis, super-resolution, and denoising algorithms [19,7,2], image patches matching a target patch are searched for repeatedly, making low-dimensional representations valuable for faster performance. PCA is a popular choice for this purpose, but may need to be applied to each example image independently for superior synthesis quality. Our fast covariance computation and low-dimensional projection algorithms significantly speed up the pre-processing time for these applications. Note that the local gist features described above can also be used in synthesis tasks for searching similar patches.

2 Related Work

Data analysis techniques such as PCA [22], LDA [6] and factor analysis [5] employ covariance matrix computation as an essential step. We specifically focus on dimensionality reduction of image patches, and the fast computation of covariance matrices for that purpose. Such efficient covariance computation would benefit several image processing applications including texture synthesis [19,17,27], image and video compression [28,20], super resolution [7,13,26], non-local denoising [2,1], inpainting [4,15], image modeling [14], and image descriptors computation [12,21].

Covariance estimation for high dimensional vectors is a classically difficult problem because the number of coefficients in the covariance grows as the dimension squared [25,8,10]. Most work on estimation of covariance matrices approximates the actual covariance matrix on the basis of a sample from a multivariate

distribution. Higham [9] provided a method for computing the nearest covariance matrix when only partially observed data are available. Cao and Bouman [3] presented a technique based on constrained maximum likelihood estimation for covariance matrices with $n < d$, where n versions of a d dimensional vector are given. We are solving the $n >> d$ case in which the observations are complete. We provide an efficient approach for the unique situation where the high dimensional vectors are sub-windows sliding in a large domain, such as from an image or an acoustic signal.

Qi and Leahy [24] described an approximate technique for fast computation of the covariance using maximum *a-posteriori* estimation. They extracted the covariance from multiple images. Porikli and Tuzel [23] presented an integral image based algorithm to efficiently extract covariance matrices from a given image. Their feature vector is composed of values defined at a single pixel. The typical dimensionality used in [23] is $d \approx 7$. On the contrary, in our method feature vectors are composed of values that span multiple pixels (patches) and have much higher dimensionality ($d = 3072$ for 32×32 RGB patches). One could express a patch based feature vector by unrolling the entire patch at every pixel and subsequently apply the integral based method for covariance computation. However, as per [23], computing the integral image takes $O(nd^2)$ time and storage (as $d + d^2$ integral images need to be computed). For large d-values, this is much slower than our method, which takes $O(n \log n)$ time. Moreover, the storage requirements for the integral image method are prohibitive in this case, requiring more than 20GB for a 100×100 image! The advantage of the integral image based method is that it allows covariance calculation over arbitrary windows in $O(d^2)$ time once the integral images have been computed. Our method on the other hand operates over the whole image (or a fixed window), but can handle an arbitrary mask or pixel weights if they are known *a-priori*.

3 Fast Covariance Computation

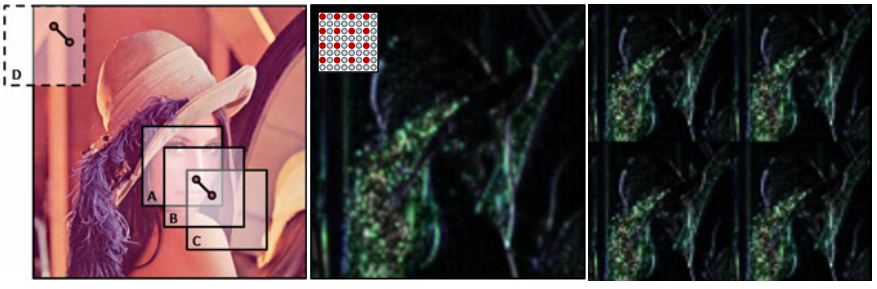
Computation of the covariance matrix from a given set of feature vectors is an expensive operation when the number and/or dimensionality of the feature vectors is large. A set of n feature vectors of dimensionality d can be expressed as the feature matrix $\mathbf{F}$:

$$\mathbf{F} = (\mathbf{f}_1 \ \mathbf{f}_2 \ \ldots \ \mathbf{f}_n), \qquad \text{where } \mathbf{f}_i = (f_{i1} \ f_{i2} \ \ldots \ f_{id})^T$$

is the i^{th} feature vector. The covariance matrix over these feature vectors (assuming zero-mean)[1] is:

$$\mathbf{C} = \frac{1}{n}\mathbf{F}\mathbf{F}^T = \frac{1}{n}\sum_{i=1}^{n} \mathbf{f}_i \mathbf{f}_i^T,$$

[1] The true covariance matrix is obtained by subtracting the outer product of the mean vector from $\mathbf{C}$.



(a) (b) (c)

Fig. 1. (a) The *diagonal pixel pair* in the middle of the image corresponds to different pixel pair locations (equivalently pairs of feature vector components) for *patches A, B, and C* (w.r.t their patch origins). Therefore the same pixel pair contributes to all pairs of feature vector component products that have the same relative pixel displacement. Pixel pairs near the boundary of the image contribute to the covariance of some "imaginary" patches (like *patch D*) that do not fully lie inside the image. (b) A filtered *local gist image* is shown along with *a sub-sampled patch*. The sub-window feature in this case is formed by collecting only *the red pixels* from the patch. Each local gist pixel stores the integrated filter response over the cell anchored at that pixel (see Section 4.1 for explanation). (c) Image obtained after repacking the local gist image.

where each term $\mathbf{f}_i\mathbf{f}_i^T$ in the summation is the outer product of the feature vector $\mathbf{f}_i$ and takes $O(d^2)$ time, leading to a total time complexity of $O(nd^2)$.

Now consider the case where the feature vectors form sub-window patches in a training image. If the patch size is, say 32×32, then for a grayscale image, the dimensionality of the feature vector is $d = 32 \times 32 = 1024$. This is quite large, given that the covariance computation varies by d^2. However, since the patches are sub-windows in an image, we can exploit the redundancy between overlapping patches to speed up the computation.

For the i^{th} image patch, its feature vector's component f_{ij} corresponds to a location in the image, say $\mathbf{q}_{ij} = (q_{ij}^x, q_{ij}^y)$. If the patch's origin is anchored at location $\mathbf{t}_i = (t_i^x, t_i^y)$ in the image, we can express this location as $\mathbf{q}_{ij} = \mathbf{t}_i + \mathbf{p}_j$, where $\mathbf{p}_j = (p_j^x, p_j^y)$ is the location expressed w.r.t the patch's origin and is the same for all patches. Therefore, we can express f_{ij} as a function of the image from which features are extracted. This could be an intensity image if we are looking at intensity features, or a processed image containing filter responses, but returns a scalar feature value as a function of the pixel location[2]. If $\mathcal{I}$ denotes the image, then

$$f_{ij} = \mathcal{I}(\mathbf{t}_i + \mathbf{p}_j). \tag{1}$$

If we focus on a single entry in the covariance matrix at (f_j, f_k), then:

[2] We consider vector-valued images in the next section.

$$\mathbf{C}(f_j, f_k) = \frac{1}{n}\sum_{i=1}^{n} f_{ij} f_{ik} \tag{2}$$

$$= \frac{1}{n}\sum_{i=1}^{n} \mathcal{I}(\mathbf{t}_i + \mathbf{p}_j)\mathcal{I}(\mathbf{t}_i + \mathbf{p}_k) \tag{3}$$

$$= \frac{1}{n}\sum_{i=1}^{n} \mathcal{I}(\mathbf{t}_i + \mathbf{p}_j)\mathcal{I}(\mathbf{t}_i + \mathbf{p}_j + \mathbf{v}_{jk}), \tag{4}$$

where $\mathbf{v}_{jk} = \mathbf{p}_k - \mathbf{p}_j$ is the displacement vector between the pixel locations corresponding to f_{ij} and f_{ik}. Now, if we treat the image as infinite, that is the number of patches $n \to \infty$ and the patch displacements $\mathbf{t}_i$ span all integer-valued locations in the plane, then we can drop $\mathbf{p}_j$ from the term $\mathbf{t}_i + \mathbf{p}_j$ in (4). This is possible because under this infinite span assumption, the pixels spanned by both $\mathbf{t}_i$ and $\mathbf{t}_i + \mathbf{p}_j$ are the same, and therefore the sum in (4) tends to the same value. Hence, we can rewrite (4) as:

$$\mathbf{C}(f_j, f_k) \approx \frac{1}{n}\sum_{i=1}^{n} \mathcal{I}(\mathbf{t}_i)\mathcal{I}(\mathbf{t}_i + \mathbf{v}_{jk}) = \mathcal{C}(\mathbf{v}_{jk}), \tag{5}$$

i.e. the covariance value is only a function of the displacement between pixel locations corresponding to the feature vector's scalar components. Intuitively, this works because the same pixel pair in the *image* contributes to the sums for different pixel pairs in different *patches*, all with the same relative displacement, as shown in Fig. 1a. In practice, for a finite sized image, this formulation results in an approximation since pixel pairs near the boundary would not contribute to all products with the same relative displacement (also shown in Fig. 1a). However, for large enough images, this is an acceptable approximation: because we are aggregating these products, the error due to the extra accumulation from boundary pixels diminishes with increasing image size.

3.1 Algorithm

To compute the covariance matrix using (5), one can compute the product for all pixel pairs in the image with the same relative displacement and sum them up. These sums of products are stored in a lookup table indexed by the relative displacement $\mathbf{v}$. The entry $\mathbf{C}(f_j, f_k)$ in the covariance matrix is then assigned as value in the lookup table at index $\mathbf{v}_{jk} = \mathbf{p}_k - \mathbf{p}_j$, where $\mathbf{p}_j$ and $\mathbf{p}_k$ are corresponding pixel locations as defined above. To analyze the complexity of this algorithm, observe that we need to do this computation for d displacement vectors because the possible integer-valued relative displacements in a $w \times w$ sized patch is $w^2 = d$ (the dimensionality of the patch feature vectors). Also, each computation is done over all pixel pairs in the image which are $O(n)$, where n is the number of pixels in the image. Therefore the total complexity is $O(nd)$. This is much better compared to the original complexity of $O(nd^2)$. For a 32×32 patch for example, this is three orders of magnitude faster.

One can further speed up covariance computation by observing that (5) represents the 2D auto-correlation function of the image $\mathcal{I}$, which can be computed efficiently in frequency domain using the Fast Fourier Transform (FFT). The complexity of this algorithm is bounded by the complexity of FFT computation, which is $O(n \log n)$. For patches with large dimensionality $d >> \log n$, this is faster than computing the lookup table by explicit summation of products.

4 Extension to Vector Images and Gist Features

The covariance computation approach described above assumes scalar-valued images. It can be extended to vector-valued images, where the feature vector is formed by concatenation of the vector components at each pixel in the patch. Vector-valued images may include multi-channel color images, or images obtained as responses of filter banks applied to the original image. For example, it is common to apply gradient or Gabor filters [11] to images for texture analysis as well as for computation of global scene features in the gist algorithm [21].

Consider a vector-valued $\mathcal{I}$ image with c channels. A feature value in an image patch now corresponds to a channel in addition to a pixel location. For the i^{th} patch, feature component f_{ij} corresponds to location $\mathbf{q}_{ij} = \mathbf{t}_i + \mathbf{p}_j$ and channel c_j. Hence, (1) and (4) respectively become

$$f_{ij} = \mathcal{I}(\mathbf{t}_i + \mathbf{p}_j, c_j), \quad \text{and}$$

$$\mathbf{C}(f_j, f_k) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{I}(\mathbf{t}_i + \mathbf{p}_j, c_j)\mathcal{I}(\mathbf{t}_i + \mathbf{p}_j + \mathbf{v}_{jk}, c_k).$$

By applying the same argument as used for deriving (5), we obtain

$$\mathbf{C}(f_j, f_k) \approx \frac{1}{n} \sum_{i=1}^{n} \mathcal{I}(\mathbf{t}_i, c_j)\mathcal{I}(\mathbf{t}_i + \mathbf{v}_{jk}, c_k),$$

i.e. , the covariance value corresponding to a pair of features is a function of the channels they belong to in addition to the relative displacement in patch space. Instead of representing the auto-correlation function of the image, the covariance now represents the cross-correlation between the respective channels of the image. Therefore, frequency domain computation can still be employed. However, the cross-correlation needs to be computed across all (unordered) pairs of image channels, making the total complexity $O(c^2 n \log n)$. However, this is still better than the complexity of the exact brute-force algorithm, which is $O(nd^2) = O(nc^2w^4)$, where w is the window size.

4.1 Sub-sampled Windows and Gist Features

We now consider sub-window features that sub-sample the original image. Figure 1b shows an example sub-sampled patch. Such sub-sampling of patches is

useful for computation of *local* gist features, which are gist features computed over patches. Compared with *global* gist features, which compute a gist image for entire image, we compute a gist patch for *every* image patch, where each cell in the gist patch is computed by integrating over a subset of pixels within the patch.

More specifically, local gist features are computed as weighted filter responses over local image patches. Firstly, a multi-channel image is obtained by applying several filters to the image such as Gabor wavelets and/or oriented gradient filters. Every patch over which the feature vector needs to be extracted is further divided into a grid of cells, where each cell contains $s \times s$ pixels (typically $s = 4$). The filtered images are integrated within these cells for each patch to form a feature vector of size $\frac{w}{s} \times \frac{w}{s} \times c$, where $\frac{w}{s}$ is the number of cells along each dimension within a patch's grid and c is the number of filtered channels. One can then organize these integrated cell responses into a *local gist image*, where each pixel stores the integrated response for the cell anchored at that pixel (see Fig. 1b for a visualization of the gist image, shown with 2×2 cells). The feature vector corresponding to a patch can then be obtained by sub-sampling the local gist image every s pixels.

These features are used to form patch-level scene descriptors in retrieval and recognition tasks. Local gist features are also useful for searching patches within an image for graphics applications such as example-based texture synthesis and super resolution.

Another application of sub-sampled patches is hierarchical processing. For example, in [18], a Gaussian stack (instead of a pyramid) is used as the multi-scale representation of an image. Patches at lower resolutions in the stack are obtained by sub-sampling from corresponding filtered images with a successively larger step size.

Covariance computation for features corresponding to such sub-sampled patches follows the observation that feature values only interact with other feature values that are a multiple of s pixels away in either dimension, where s is the sub-sampling step size. Therefore one can *re-pack* the image pixels so that it results in a grid of $s \times s$ sub-images (as shown in Fig. 1c), where each sub-image now consists of densely sampled $\frac{w}{s} \times \frac{w}{s}$ patches. Covariance matrices may then be computed independently for each of these sub-images and averaged together to obtain the combined covariance. Alternately, because the sub-images need to be processed independently, there is no performance benefit to processing all of them together (as was the case with processing all patches together). Hence, it may be sufficient to compute the covariance based on just one of the sub-sampled images. Since each sub-image contains $\frac{n}{s^2}$ pixels, the complexity is $O(c^2 \frac{n}{s^2} \log \frac{n}{s^2})$ per sub-image (or $O(c^2 n \log \frac{n}{s^2})$ if all sub-images are used).

The re-packing described above may also be used for processing multiple images simultaneously by concatenating them together into a larger collage if the number of images is small. Alternatively, covariances for each image can be computed independently followed by weighted averaging.

5 Weighted Features

The above approach for covariance computation can be extended to the case in which pixels have arbitrary weights. This may be useful in case certain patches are more preferrable than others, *e.g.* those near interest points or high gradients. The caveat is that the weights need to be expressed per-pixel, as opposed to per-patch. However, a simple way to achieve that is to assign to every pixel the average weight of patches overlapping it. The per-pixel weights may also be used to specify an image mask that selects the pixels to be considered. In presence of weights, (5) becomes

$$\begin{aligned}\mathbf{C}(f_j, f_k) &\approx \frac{\sum_{i=1}^{n} \mathbf{W}(\mathbf{t}_i)\mathcal{I}(\mathbf{t}_i)\mathbf{W}(\mathbf{t}_i + \mathbf{v}_{jk})\mathcal{I}(\mathbf{t}_i + \mathbf{v}_{jk})}{\sum_{i=1}^{n} \mathbf{W}(\mathbf{t}_i)\mathbf{W}(\mathbf{t}_i + \mathbf{v}_{jk})} \\ &= \frac{\sum_{i=1}^{n} \mathbf{W}\mathcal{I}(\mathbf{t}_i)\mathbf{W}\mathcal{I}(\mathbf{t}_i + \mathbf{v}_{jk})}{\sum_{i=1}^{n} \mathbf{W}(\mathbf{t}_i)\mathbf{W}(\mathbf{t}_i + \mathbf{v}_{jk})}\end{aligned}$$

where $\mathbf{W}$ denotes the per-pixel weights and $\mathbf{W}\mathcal{I}$ denotes the *weighted image*, obtained by multiplying the weights with the image at every pixel. The numerator and denominator denote cross-correlation and auto-correlation operations respectively and therefore can be computed efficiently as described earlier.

6 Fast Dimensionality Reduction

The covariance matrix computation described above can be used as a pre-process for performing PCA or LDA on the original feature vectors. However, to use the computed principal components for dimensionality reduction, it is necessary to project the original high-dimensional feature vectors onto the low-dimensional space represented by the principal basis. We can again exploit the redundancy across overlapping sub-windows to perform this operation efficiently as well.

Projecting a sub-window patch onto a single principal basis vector entails computing a dot product between the two vectors which is an $O(d)$ operation, where $d = c \times w \times w$ is the dimensionality of the patch. Therefore projecting *all* sub-windows within the image onto a single basis vector requires $O(nd) = O(ncw^2)$ computation for an image with n pixels. However, if we interpret each principal component vector as a patch, then the basis coefficient b_k for an image patch anchored at location $\mathbf{t}_i$ w.r.t the k^{th} principal basis patch $\mathcal{B}_k$ can be expressed as:

$$b_k(\mathbf{t}_i) = \sum_{l=1}^{c} \sum_{j=1}^{w^2} \mathcal{I}(\mathbf{t}_i + \mathbf{p}_j, c_l)\mathcal{B}_k(\mathbf{p}_j, c_l)$$

where $\mathbf{p}_j$ spans the $w \times w$ patch window. Since we want to compute b_k for all values of $\mathbf{t}_i$, this is equivalent to filtering the image $\mathcal{I}$ with the basis patch $\mathcal{B}_k$. This can be again efficiently computed in $O(cn \log n)$ time in the frequency domain, which is significantly faster when the patch size is non-trivial, *i.e.* $w^2 >> \log n$.

7 Experimental Results

In our experiments, we compute covariances for patches extracted from RGB images and gist images (which are 6 channel Gabor-filtered images with responses integrated over cells of 4×4 pixels). We have used a dataset consisting of texture images, natural scenes, and urban imagery (see Fig. 1a and Fig. 2 for a few examples that we will refer to subsequently). We compute covariances using three methods: (1) the *exact* method that computes the average covariance over all feature vectors explicitly, (2) our frequency domain *FFT*-based method, and (3) a *sampling* method that sub-samples the image for feature vectors, only using $\frac{n}{w^2}$ vectors, either randomly or over a regular grid.

Table 1 demonstrates the performance gain we achieve in covariance computation as well as PCA projection over the respective exact methods. Our covariance computation is 2-3 orders of magnitude faster, while projection is about an order of magnitude faster. The sampling method is slower than ours for the chosen sampling rate, without being as accurate. Random sampling or sampling over a grid generate similar results. Figure 3 shows a visualization of the covariance matrices and the principal components obtained using the three methods for the Crowd image. The covariance matrix obtained by our method has the same

(a) Crowd (b) Traffic (c) Leopard skin (d) Green scales

Fig. 2. Reference images used in quantitative experiments (also refer Lena in Fig. 1a)

(a) (b)

Fig. 3. Comparison of covariance matrices and PCA basis vectors computed over 16×16 patches extracted from the Crowd image. Visualization of resulting *covariance matrices* (a) and *top 25 basis vectors* (b): shown from left-to-right are the results for the exact method, our FFT-based method, and the sampling method, respectively. The covariance matrix obtained by our method has the same structure as the exact covariance, while the sampling method exhibits aliasing which is also evident in the principal components. The principal components obtained by our method closely resemble the smooth exact bases, with the top few components being nearly identical.

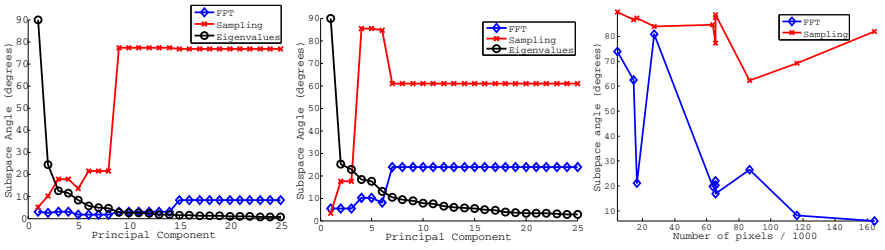
Table 1. Performance comparison between various methods. CPU time shown in seconds. Our method (FFT) is 2-3 orders of magnitude faster for covariance computation, and about an order of magnitude faster for projection, compared with exact method. The sampling method is slower than ours for the chosen sampling rate.

Image (size)	Covariance Time						Projection Time	
	16×16 RGB			32×32 gist			16×16 RGB	
	Exact	**FFT**	Sampling	Exact	**FFT**	Sampling	Exact	**FFT**
Scales (64x64)	6	0.04	0.04	0.62	0.02	0.02	0.7	0.1
Grass (120x120)	24	0.07	0.12	5	0.06	0.08	3	0.5
Leopard (128x128)	29	0.07	0.15	5	0.06	0.1	3.5	0.6
Crowd (150x180)	50	0.11	0.23	10	0.09	0.17	6	1.2
Gecko (256x256)	130	0.23	0.57	29	0.23	0.5	16	3
Lena (256x256)	130	0.23	0.59	29	0.22	0.49	16	3
Text (256x256)	131	0.23	0.57	29	0.25	0.49	16	3.2
Windows (306x208)	125	0.23	0.57	27	0.22	0.45	16	3.6
Ropes (360x240)	177	0.3	0.74	39	0.32	0.64	21	5
Traffic (390x300)	239	0.41	0.97	54	0.42	0.85	29	7.5
Building (865x190)	341	0.6	1.34	74	0.74	1.22	41	8

structure as the exact covariance, while the sampling method exhibits aliasing as it is biased towards the sampled patches (also evident in the principal components). Note that the aliasing is not due to sampling on a regular grid since contributions from all samples are averaged together. The principal components obtained by our method, on the other hand, closely resemble the smooth exact bases, with the top few components being nearly identical.

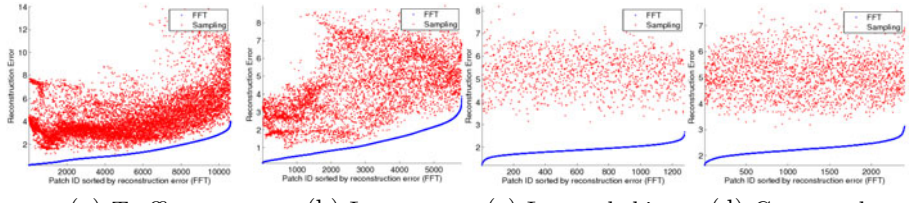
Figure 4 is a quantitative comparison of the covariance matrices computed using our method and the sampling method against the exact covariance. Since we are ultimately interested in the principal components obtained from the covariance, we compare the subspaces induced by these components. We group successive principal components obtained from the exact method into subspaces if the ratio between their respective eigenvalues is less than a threshold (1.2 in our experiments). This is necessary because the principal eigenvectors become unstable when their corresponding eigenvalues are close to each other. Therefore it makes more sense to compare the grouped subspaces as opposed to individual eigenvectors. Note that we do consistently better than the sampling method (*i.e.* have smaller subspace angles). Also, the subspace angle increases only close to where the eigenvalue curve becomes flat, *i.e.* after most of the variance has been captured. The rightmost plot shows that the subspace angle for our method generally decreases as the number of pixels in the image increase, confirming our hypothesis that the approximation should improve with image size.

Figure 5 compares the reconstruction error and Fig. 6 compares the nearest neighbor (NN) search performance between our method (FFT) and sampling method, Our method results in lower reconstruction error, and the NNs from our method consistently have lower true distances to query patches than sampling method. The details are in the captions of Fig. 5 and Fig. 6.



(a) (b) (c)

Fig. 4. Subspace angles between principal components obtained by each method (FFT and sampling) w.r.t the exact method. (a) is a plot for 16×16 patches extracted from an RGB image (Traffic), and (b) is a plot for 32×32 patches in a 6-channel gist image (corresponds to an 8×8 grid of cells 4×4 pixels each). In these two plots, the eigenvalues of principal components are plotted (scaled and shifted to fit graph). Our method consistently generates smaller subspace angles compared with sampling method. There is a jump in the curve when a new subspace is created, and the curve stays flat when the new eigenvector is added to the same subspace but does not change the angle considerably. (c) shows the angle between the subspaces induced by the top 10 eigenvectors of the two methods for different images listed in Table 1, plotted as a function of number of pixels in the images. The subspace angle for our method generally decreases as the number of pixels in the image increase, confirming our hypothesis that the approximation should improve with image size.



(a) Traffic (b) Lena (c) Leopard skin (d) Green scales

Fig. 5. Comparison of reconstruction error (*y-axis*) based on top 25 PCA coefficients for basis vectors computed using our FFT method (*blue curve*) and the sampling method (*red dots*). Each plot shows reconstruction error for all 16×16 patches in the image, sorted (along the *x-axis*) by increasing FFT method error. The error from our method's PCA basis is consistently smaller than that from the sampling method.

We have applied our technique for accelerating example-based super resolution and texture synthesis. A practical setting where these methods may be employed is for resolution enhancement and hole filling of building facades in large scale 3D urban environments. For super resolution, given a low resolution target image and a high resolution (partial) source image with similar texture, we synthesize

a high resolution version of the target (our algorithm combines ideas from [7] and [16]). The core of the synthesis algorithm is patch-based search performed in PCA space. The most time consuming component is covariance computation and feature projection to PCA space. Figure 7a shows a sample super resolution result. We extract 32×32 patches in this 202×402 image. The computation time is 3.1s using the fast covariance computation vs. 2415s using the exact covariance computation. A similar approach may be used for hole filling (see Fig. 7b). Again, we take advantage of the fast covariance computation to improve the processing time by more than 2 orders of magnitude.

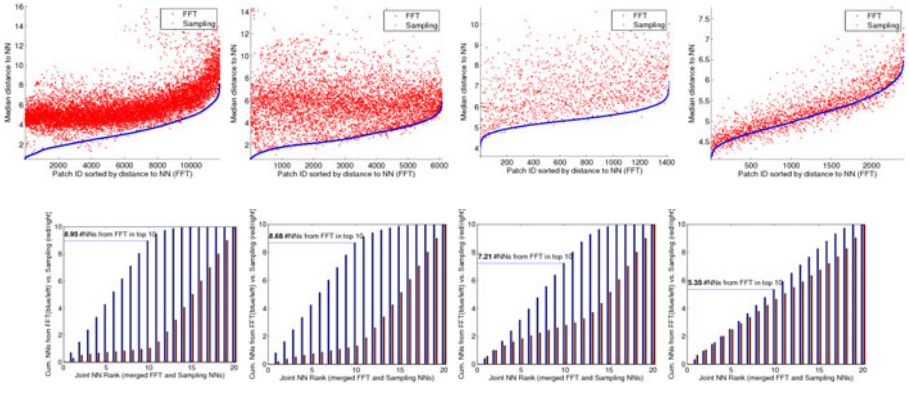


Fig. 6. Nearest neighbor performance on 16×16 patches from (left to right) Traffic, Lena, Leopard skin and Green scales. Top row plots the median distance (*y-axis*; *blue curve* for our FFT method and *red dots* for sampling method) from every patch to its top 10 nearest neighbors. Distances are computed over PCA coefficients. We use a hierarchical search tree constructed from PCA projected patches for nearest neighbor search. Patches are sorted (along the *x-axis*) by FFT method distance. These plots demonstrate that our FFT method consistently results in nearest neighbors with smaller median distance compared to the sampling method, except for Green scales where performance is more even. This is attributable to the fact that the Green scales texture is small in size (64×64) and therefore the approximation error in covariance matrix computation is not negligible. The bottom row plots cumulative histograms over the joint rank of nearest neighbors collected from the two methods (FFT and sampling). We find the top 10 nearest neighbors for each patch from both methods and jointly ranked the resulting 20 neighbors. Then for top K neighbors where K varies from 1 to 20 (plotted along the *x-axis*), we count how many neighbors come from the FFT method (*blue left-side bars*) vs. the sampling method (*red right-side bars*) and average this value over all patches. The resulting value (plotted on the *y-axis*) denotes the average number of nearest neighbors in the top-K, that come from the FFT method. Note that for the first three columns, this value for $K = 10$ lies between 7 and 9, indicating that the FFT method consistently results in better ranked neighbors. For the last column (Green scales), the performance is again evenly split (5.35) between the two methods.

(a) (b)

Fig. 7. Sample Applications: (a) Example-based super resolution of building facades. Left to right: (partial) high resolution source image, low resolution target image, high resolution result. (b) Hole filling. The texture on the left contains a hole (shown in *black*) which is filled on the right using texture synthesis.

8 Conclusion

We have proposed a novel algorithm to efficiently compute covariance matrices for features that can be described as sub-windows in an image. The overlapping nature of these sub-windows results in a special property for the covariance matrix, namely that the covariance between two pixel features is a function of their relative displacement. Using this property, covariance computation can be expressed as a cross-correlation operation, which can be computed quickly in the frequency domain. We have also presented extensions for vector-valued images and sub-sampled windows, as well as a method for fast low-dimensional projection of the sub-windows onto PCA space. Our formulation results in an approximation to the exact covariance, where the approximation error diminishes with increasing image size. We support this claim with both qualitative and quantitative experimental results. We also compare with a simple sampling approach to covariance estimation, and show that our technique results in a much closer approximation, while still being faster.

References

1. Adams, A., Gelfand, N., Dolson, J., Levoy, M.: Gaussian kd-trees for fast high-dimensional filtering. ACM Trans. Graph., SIGGRAPH 28(3), 1–12 (2009)
2. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 60–65. IEEE Computer Society Press, Los Alamitos (2005)
3. Cao, G., Bouman, C.A.: Covariance estimation for high dimensional data vectors using the sparse matrix transform. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (eds.) NIPS, pp. 225–232. MIT Press, Cambridge (2008)
4. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. IEEE Transactions on Image Processing 13 (2004)
5. Darlington, R.B., Weinberg, S., Herbert, W.: Canonical variate analysis and related techniques. Review of Educational Research, 453–454 (1973)

6. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of Eugenics 7, 179–188 (1936)
7. Freeman, W.T., Jones, T.R., Pasztor, E.C.: Example-based super resolution. IEEE Comput. Graph. Appl. (2002)
8. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Computer Science and Scientific Computing Series. Academic Press, London (1990)
9. Higham, N.J.: Computing the nearest correlation matrix a problem from finance. IMA Journal of Numerical Analysis 22(3), 329–343 (2002)
10. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(1), 4–37 (2000)
11. Jones, J.P., Palmer, L.A.: An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. J Neurophysiol 58(6), 1233–1258 (1987)
12. Ke, Y., Sukthankar, R.: Pca-sift: a more distinctive representation for local image descriptors. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 506–513 (2004)
13. Kim, K.I., Franz, M., Schölkopf, B.: Kernel hebbian algorithm for single-frame super-resolution. In: Leonardis, A., Bischof, H. (eds.) Statistical Learning in Computer Vision, pp. 135–149. Springer, Berlin (2004)
14. Kim, K.I., Franz, M.O., Schölkopf, B.: Iterative kernel principal component analysis for image modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(9), 1351–1366 (2005)
15. Korah, T., Rasmussen, C.: Pca-based recognition for efficient inpainting. In: IEEE Asian Conference on Computer Vision (2006)
16. Kwatra, V., Essa, I., Bobick, A., Kwatra, N.: Texture optimization for example-based synthesis. ACM Trans. Graph., SIGGRAPH 24(3), 795–802 (2005)
17. Lefebvre, S., Hoppe, H.: Appearance-space texture synthesis. In: Proc. of SIGGRAPH 2006, pp. 541–548 (2006)
18. Lefebvre, S., Hoppe, H.: Parallel controllable texture synthesis. ACM Transactions on Graphics, SIGGRAPH, 777–786 (2005)
19. Liang, L., Liu, C., Xu, Y., Guo, B., Shum, H.Y.: Real-time texture synthesis by patch-based sampling. ACM Trans. Graph. 20(3), 127–150 (2001)
20. Liu, J., Wu, F., Yao, L., Zhuang, Y.: A prediction error compression method with tensor-pca in video coding. In: MACM, pp. 493–500 (2007)
21. Oliva, A., Torralba, A.: Building the gist of a scene: the role of global image features in recognition. Progress in Brain Research 155, 23–36 (2006)
22. Pearson, K.: On lines and planes of closest fit to systems of points in space. Philosophical Magazine 2(6), 559–572 (1901)
23. Porikli, W.F., Tuzel, O.: Fast construction of covariance matrices for arbitrary size image. In: Proc. Intl. Conf. on Image Processing, pp. 1581–1584 (2006)
24. Qi, J., Leahy, R.M.: Fast computation of the covariance of map reconstructions of pet images. Proceedings of SPIE 3661(1), 344–355 (1999)
25. Stein, C., Efron, B., Morris, C.: Improving the usual estimator of a normal covariance matrix. Dept. of Statistics, Stanford University, Report 37 (1972)
26. Wang, Q., Tang, X., Shum, H.Y.: Patch based blind image super resolution. In: ICCV (2005)
27. Wei, L.Y., Lefebvre, S., Kwatra, V., Turk, G.: State of the art in example-based texture synthesis. In: Eurographics 2009, State of the Art Report, EG-STAR. Eurographics Association (2009)
28. Yu, Y.D., Kang, D.S., Kim, D.: Color image compression based on vector quantization using pca and lebld. In: Proc. of the IEEE Region 10 Conference, vol. 2, pp. 1259–1262 (1999)

Binary Coherent Edge Descriptors

C. Lawrence Zitnick

Microsoft Research, Redmond, WA

Abstract. Patch descriptors are used for a variety of tasks ranging from finding corresponding points across images, to describing object category parts. In this paper, we propose an image patch descriptor based on edge position, orientation and local linear length. Unlike previous works using histograms of gradients, our descriptor does not encode relative gradient magnitudes. Our approach locally normalizes the patch gradients to remove relative gradient information, followed by orientation dependent binning. Finally, the edge histogram is binarized to encode edge locations, orientations and lengths. Two additional extensions are proposed for fast PCA dimensionality reduction, and a min-hash approach for fast patch retrieval. Our algorithm produces state-of-the-art results on previously published object instance patch data sets, as well as a new patch data set modeling intra-category appearance variations.

1 Introduction

The ability to describe an image patch is critical to many recognition algorithms. Image patches can be used to find correspondences between varying viewpoints of an object [1,2,3,4], or to represent parts of object categories [5,6,7]. Typically, a desirable patch descriptor is robust to illumination changes, moderate pose variation, and intra-category appearance variation.

A standard approach to describe a patch is the use of Histograms of Gradients (HoG), [1,7,8,9,10,11]. A HoG is defined as the histogram of image gradients over a combination of positions, orientations and scales. Examples include the SIFT [1] and GLOH [9] interest point descriptors, which have been shown to be very effective for object instance recognition. Similar approaches have been applied to describe object category parts [7,12]. After creating histograms from local pixel gradients, standard HoG approaches rely on a global normalization step to account for variations in illumination. However, these descriptors are still sensitive to the relative magnitudes of gradients. In many scenarios such as intra-category appearance variation and partial illumination changes the relative gradient magnitudes do vary, resulting in reduced matching performance. Several approaches [1,11,12] use truncated normalization to help reduce this sensitivity.

In this paper, we propose an image patch descriptor based on the location, orientation, and length of edges, and not their relative gradient magnitudes. We hypothesize that the presence and not magnitude of edges provides an informative measure of patch similarity that is robust not only to illumination and pose changes, but intra-category appearance variation. Our descriptor encodes the

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 170–182, 2010.

presence or absence of edges using a binary value for a range of possible edge positions and orientations. In addition the locally linear length of an edge is used to differentiate sets of coherent edges aligned perpendicular to the edge orientation from shorter edges resulting from textures. Our approach consists of three main steps: First, the image patch gradients are locally normalized to remove variations in relative gradient magnitudes. Second, the normalized gradients are binned using the position, orientation and local linear length of an edge. Finally, the normalized gradient histogram is binarized to encode the presence of edges.

In addition to the basic approach we propose two extensions: a fast method for dimensionally reduction using binary vectors and PCA, and a min-hash feature representation for efficient retrieval. The approach is tested using a previously published [11] ground truth object instance data set to test its invariance to illumination and pose changes. A new data set is provided to test invariance to intra-category appearance variation. In both cases, state-of-the-art results are achieved, with significant increases in accuracy over traditional approaches such as SIFT [1], GLOH [9] and variants of Daisy descriptors [10,11].

The rest of the paper is organized as follows: In the next section we describe previous work, followed by our basic approach. In Section 4 we discuss extensions to our algorithm. Finally results are provided in Section 5 following by a conclusion and discussion.

2 Previous Work

There exists a large body of previous work on image patch descriptors [13]. The SIFT [1] descriptor popularized the HoG approach and introduced several optimizations, including truncated normalization and ratio tests. Several follow up papers have improved on the SIFT desciptor using PCA [14], radial binning [9] and "daisy" binning [11,15]. Spatial binning parameters have also been learned from training data [10,11]. Geometric Blurring [8] proposed blurring the gradients using a spatially varying blur kernel based on the distance to the center of the patch. SURF [16] uses Harr wavelets instead of gradients to describe image patches. Another approach is to use generative models to learn the statistics of image patches [17].

Image patches have also been described and classified using randomized trees [18,19] and boosting [20] to aid in detecting object classes.

Gradients are commonly used for category part representation. Felzenszwalb et al. [7] and Dalal and Triggs [12] use HoGs for object category detection, while others such as Crandall et al. [6] use binary edge detection. PCA on image intensities has also shown good results in Fergus et al. [5].

3 Binary Edge Descriptor

Our descriptor relies on the detection of edges in an image patch. It is assumed that the presence of edges remains consistent across matching image patches, even if their relative magnitudes do not. Thus, we describe an edge based on its

orientation, position and length, and not its gradient magnitude. For instance see Figure 1(a). Both patches share similar edge structure, but the relative gradient magnitudes vary significantly.

Our method is split into three stages: gradient normalization, edge aggregation and binarization. Gradient normalization removes differences in relative gradient magnitudes between edges. It is worth noting that we locally normalize gradients to remove relative differences in magnitude, instead of a global normalization [1] that only accounts for global gain and offset differences. Next, the gradients are aggregated into bins, after which a binarization process labels the bins with highest contribution. Before we describe these three stages, we define our initial gradient orientations and magnitudes.

The descriptor is computed from a $n \times n$ square patch of pixels. The intensity of a pixel p at location (x_p, y_p) is denoted $f(p)$ or $f(x_p, y_p)$. The horizontal gradient $f_x(p)$ of the pixel is equal to $f(x_p+1, y_p)-f(x_p, y_p)$ and similarly for the vertical gradient $f_y(p)$. The magnitude of the gradient for pixel p is the Euclidean norm of its gradients, $g_p = \|[f_x(p)\ f_y(p)]^{\mathrm{T}}\|_2$. The orientation is defined as $\theta_p = \arctan(f_y(p)/f_x(p))$. To help remove noise and sampling artifacts a small amount of Gaussian blur ($\sigma = 0.5$) is applied to the patch before computing the gradients and orientations.

3.1 Gradient Magnitude Normalization

Our goal for gradient normalization is to maintain the gradient profiles while removing the relative height differences between the gradient peaks. An efficient method to solve this problem is to normalize the gradients based on the average gradient magnitude in a local spatial neighborhood. We compute the average Gaussian weighted gradient magnitude $\bar{g}_p$ in a spatial neighborhood N of p using

$$\bar{g}_p = \sum_{q \in N} g_q \mathcal{N}(q; p, \sigma_s), \tag{1}$$

where $\mathcal{N}$ is the standard normal distribution. The normalized gradients $\hat{g}_p$ are computed using the ratio of the original gradients and the average gradients,

$$\hat{g}_p = \frac{g_p}{\max(\bar{g}_p, \epsilon)}, \tag{2}$$

where $\epsilon = 4$ is used to ensure the magnitude of $\bar{g}_p$ is above the level of noise. In our experiments the spatial standard deviation is set to $\sigma_s = 3$. Examples of the normalized gradients are shown in Figure 1(a). We also experimented with including orientation to compute the average gradients in three dimensions. This avoids edges with large gradient magnitudes inhibiting the gradients of nearby edges with different orientations. However, this computationally more expensive approach did not improve the accuracy of the final descriptor.

3.2 Edge Aggregation

The next stage of our approach aggregates the normalized gradients into bins defined by an edge's position, orientation and local linear length. We align the

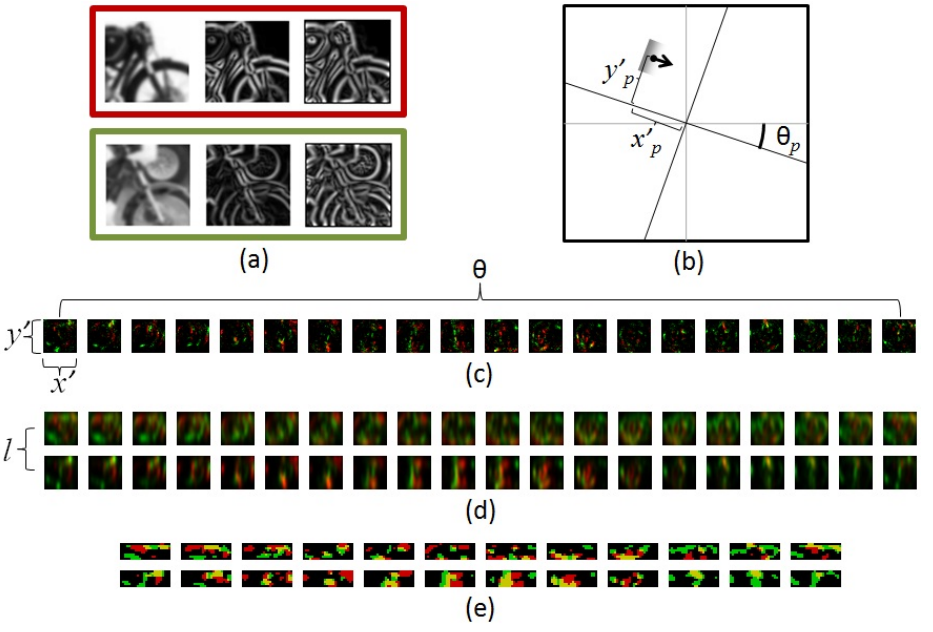


Fig. 1. Processing pipeline: (a) Two matching patches (left to right): Original patch, gradients g_p and normalized gradients $\hat{g}_p$, (b) an illustration of the coordinate frame used for the orientation dependent binning, (c) the edge histogram with the top patch shown in red and bottom patch shown in green (yellow denotes agreement), (d) the histogram after blurring and spliting the edges into two sets of bins based on the edge's local linear length, (e) final binarized descriptor, red is the top patch, green is the bottom patch and yellow denotes agreement

spatial binning with the gradient's orientation to allow for the descriptor's robustness to vary perpendicular and parallel to an edge. Orientation dependent sampling also aids in the detection of coherent edges (as shown later), i.e. sets of similarly orientated gradients aligned perpendicular to the gradient orientation. This varies from previous approaches [1,9,11] that define the spatial binning independent of the orientation. Specifically as illustrated in Figure 1(b), we define a new coordinate frame (x'_p, y'_p) for each pixel p at position (x_p, y_p) depending on its orientation θ_p equal to

$$\begin{bmatrix} x'_p \\ y'_p \end{bmatrix} = \mathbf{R}(\theta_p) \begin{bmatrix} x_p \\ y_p \end{bmatrix}, \tag{3}$$

where $\mathbf{R}(\theta_p)$ is a standard 2D rotation matrix. We assume the origin $(0, 0)$ is at the center of the patch. Using (x'_p, y'_p, θ_p) we define our binning on a $b_{x'} \times b_{y'} \times b_\theta$ resolution grid creating a histogram $H(x', y', \theta)$. In practice we use $b_{x'} = 32$, $b_{y'} = 32$ and $b_\theta = 20$. When assigning the values $\hat{g}_p$ to each bin according to (x'_p, y'_p, θ_p), we use the standard linear soft binning approach using

bilinear interpolation [1]. An example of the resulting bin values can be seen in Figure 1(c).

Detecting coherent edges: Above, we aggregated the normalized gradients into a fixed number of bins in a 3D Histogram. Specifically, we split the vertical y' dimension into $b_{y'}$ bins, capturing edges $1/b_{y'}$ the length of the patch. Many edges run the entire length of the patch. The discriminability of the descriptor could be increased if long coherent edges could be distinguished from shorter texture edges. A simple approach to estimate edge length $L(x', \theta)$ for an edge at position x' and orientation θ is to sum the vertical bins perpendicular to its gradient's direction,

$$L(x', \theta) = \sum_{y'} H(x', y', \theta). \tag{4}$$

If we assign a value of $l_p = L(x', \theta)$ to every gradient $\hat{g}_p$ we may create a four dimensional histogram $H(x', y', \theta, l)$. In our experiments we found discretizing the edge lengths into two bins, $b_l = 2$, results in an effective separation of coherent edge gradients and short texture edges, as shown in Figure 1(d). Specifically, we compute a delta function $\Delta(l_p)$ equal to

$$\Delta(l_p) = \max(0, \min(1, \frac{l_p - \alpha}{\beta})). \tag{5}$$

where the values α and β where set to 2 and 8 respectively. Other sigmoid functions may also be used, but this linear form provides efficient computation. The normalized gradient values $\hat{g}_p$ are split between the two edge length bins using $\Delta(l_p)$ and $1 - \Delta(l_p)$ as weights.

3.3 Binary Representation

Given a 4D histogram $H(x', y', \theta, l)$ we want to determine the edges present in the patch, while providing robustness to small changes in position and orientation. Robustness is provided by applying a small amount of blur to the histogram. We apply Gaussian blurring in the x', y' and θ dimensions with standard deviations of $\sigma_{x'}$, $\sigma_{y'}$ and σ_θ respectively. Optimizing over possible values of $\sigma_{x'}$, $\sigma_{y'}$ and σ_θ we empirically found values of $\sigma_{x'} = 1$, $\sigma_{x'} = 3$ and $\sigma_\theta = 1$ to work well. An increased amount of blur is applied to the $\sigma_{y'}$ dimension parallel to the edges, since this dimension proved less informative in our experiments, see Section 5. An example of the blurred histogram is shown in Figure 1(d).

Before binarizing edges in the histogram, we first reduce its resolution to $n_{x'} \times n_{y'} \times n_\theta \times n_l$ using sub-sampling. Empirically we found dimensions of $n_{x'} = 24$, $n_{y'} = 8$, $n_\theta = 12$, and $n_l = 2$ for the x', y', θ and l dimensions respectively to provide good results. Experiments for various values of $n_{x'}$, $n_{y'}$, n_θ, n_l and are shown in Section 5.

We binarize the sub-sampled histogram's values by assigning a value of 1 to the top τ percent of the bins with highest values, and 0 to the others. To reduce bias in the detection of longer edges over texture edges, we perform binarization

independently for both sets of edge length bins. The final binarized descriptor is denoted D, and an example is shown in Figure 1(e). The binarization process provides nearly full invariance to edge magnitudes. It also provides computational advantages when reducing the descriptor's dimensionality as we discuss in Section 4.1. In our experiments $\tau = 20\%$. Results using other values are shown in Section 5. In practice, several efficient $O(n)$ methods for finding the top τ percent may be used and are commonly referred to as "selection algorithms" [21].

4 Extensions

In this section, we describe two separate extensions to our basic approach for reducing the dimensionality of our descriptor using PCA and min-hash.

4.1 Dimensionally Reduction Using PCA

The size of our descriptor D is $n_{x'} \times n_{y'} \times n_\theta \times n_l$, which for the values described above is 4,608 dimensions. This is far larger than standard descriptors such as SIFT using 128 dimensions. The difference isn't quite as dramatic if it is considered that our descriptors are binary. For instance, we could store our descriptor in the same space as 144 32-bit floating point numbers. Furthermore, comparison between descriptors can be done efficiently using bit-wise *xor* functions [22,23,24].

In this section we explore dimensionally reduction using Principal Component Analysis (PCA). It has been shown [9,11,14] that using PCA can both decrease the dimensionally of a descriptor and improve accuracy. We perform PCA using a standard approach to compute K basis vectors. The training dataset Yosemite provided by [11] was used to learn the basis functions. Using real-valued descriptors, the cost of projecting an M dimensional descriptor using K basis functions uses MK multiplications and additions, which can be computationally expensive for large descriptors.

To increase efficiency, we can take advantage of two properties of our descriptors; they are binary and neighboring values typically have the same values, Figure 1(e). As a result, we can use a technique similar to integral images to efficiently project our descriptors by pre-computing the following values

$$\mathrm{w}^{\Sigma}_{k,i} = \sum_{j<i} \mathrm{w}_{k,j}, \tag{6}$$

where $\mathrm{w}_{k,i}$ is the ith value in the kth basis vector. Thus, $\mathrm{w}^{\Sigma}_{k,i}$ is the sum of all values in w_k before the ith entry. To compute the reduced dimensional descriptor D^* the kth projection of D is computed as

$$D^*_k = \sum_i (D_{i-1} - D_i)\mathrm{w}^{\Sigma}_{k,i}. \tag{7}$$

Since $(D_{i-1} - D_i)$ is only nonzero when neighboring values aren't equal, the total amount of computation is greatly reduced. In our experiments, on average

only 10% of neighboring values were not equal when parsing the descriptor using an x', y', θ and l ordering of the dimensions, resulting in just $0.1*MK$ adds on average to project onto the PCA vectors. To handle boundary conditions, an additional entry has to be added to the end of all descriptors with a value of 0. Results using PCA dimensionality reduction can be found in Section 5.

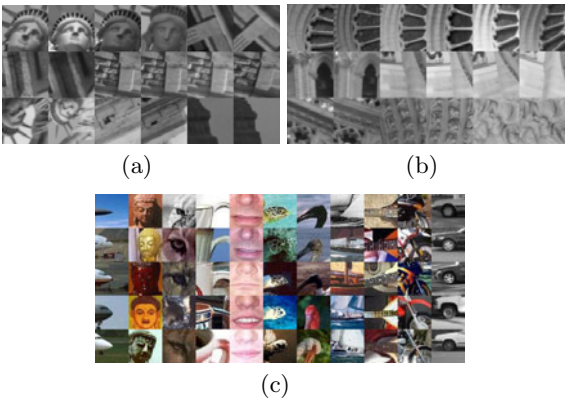
4.2 Min-hash for Fast Patch Retrieval

We propose using min-hash as an efficient means for finding similar descriptors. Previous works use min-hash [25] for image retrieval and clustering [26,27]. Locality sensitive hashing, semantic hashing and binary coding [28,23,24] have also been used for image retrieval. Hashing techniques used in conjunction with inverse look-up tables provide a fast and scalable method for finding similar points in high dimensional spaces with certain probabilistic guarantees. In particular, the min-hash technique has the property that the probability of two hashes being identical is equal to the Jaccard similarity. The Jaccard similarity is the cardinality of the intersection of two sets divided by their union's cardinality. In our task, the elements of the set are the indices assigned to 1 by our descriptor. A min-hash is found by creating a random permutation of the set of possible indices. The smallest permutated index with a value of one in a descriptor is its resulting hash value [25]. Multiple hashes can be generated for a single descriptor using different random permutations. Given a set of descriptors with hashes, an inverse lookup table can be created to efficiently find descriptors with equal hash values. If enough hashes are shared between two descriptors, they are said to "match". The advantage of hashing over simple quantization such as vocabulary trees [2] and kd-trees is the matching accuracy is proportional to the number of hashes stored per descriptor and not fixed based on the amount of quantization. In this regard it is similar to using randomized kd-trees [29] or multiple quantizations, except the quantized values can be efficiently computed without traversing a tree.

In order to increase the uniqueness of a hash, hashes can be concatenated into sketches. The size of the sketch refers to the number of hashes used to create it. If the Jaccard similarity between two patches f and f' is $J(f, f')$, the probability of two sketches being identical is $J(f, f')^k$, where k is the sketch size. Min-hash is increasingly effective if the Jaccard similarity between matching images is high and is low for non-matches. In Figure 4(a), we see the density functions for matching and non-matching image pairs with respect to the Jaccard similarity. Since our descriptor produces significant separation between the two distributions and it is binary, it is a good candidate for the min-hash algorithm. We present results using the min-hash approach with various sketch sizes and numbers of sketches in Section 5.

5 Experimental Results

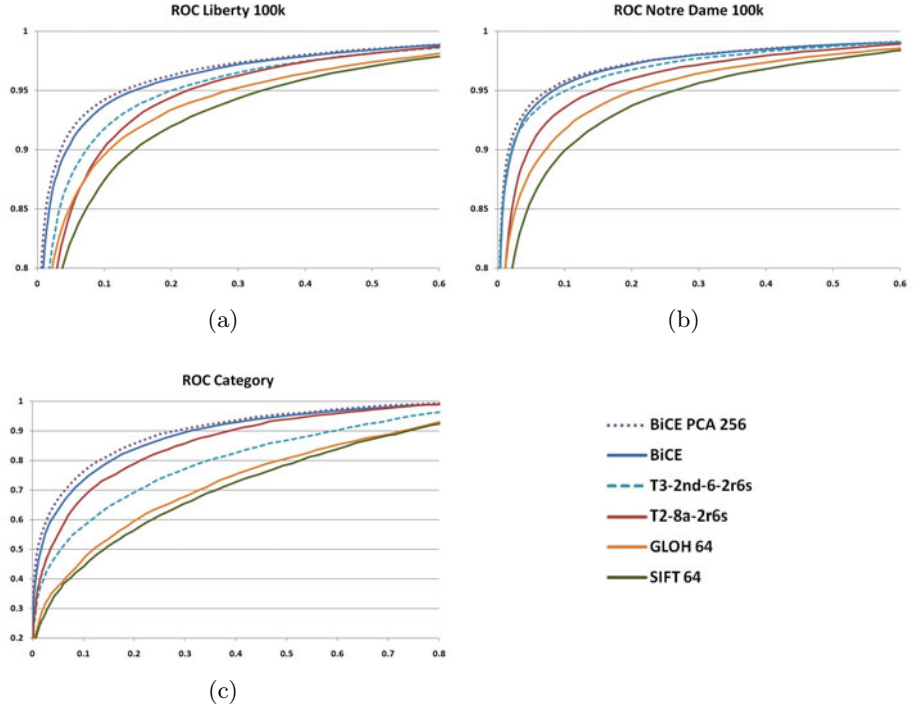
In this section, we provide experimental results on three datasets. The Liberty and Notre Dame datasets [11] contain image patches generated from Difference of



(a) (b)

(c)

Fig. 2. Examples of matching patches from the (a) Liberty, (b) Notre Dame and (c) category datasets



(a) (b)

(c)

Fig. 3. ROC curves for (a) Liberty, (b) Notre Dame and (c) Category datasets. Notice the plotted ranges vary from (a,b) to the more difficult dataset of (c).

Table 1. Liberty, Notre Dame and Category dataset accuracies for SIFT [1], Gloh [9], T2-8a-2r6s, T3-2nd-6, and T3-2nd-6 [11] compared to our approach BiCE. Errors at 95 % recall, and Equal Error Rates (EER) are given. (B) indicates binary dimensions.

		Liberty		Notre Dame		Category	
Method	Dimensions	95% Error	EER	95% Error	EER	95% Error	EER
SIFT [11]	128	35.09	-	26.10	-	-	-
SIFT 64	128	33.38	11.51	26.31	10.03	87.37	32.53
GLOH 64	272	28.38	10.27	20.46	8.87	86.69	31.25
T2-8a-2r6s	104	22.37	9.89	14.70	7.57	54.49	20.59
T3-2nd-4 [11]	416	19.36	-	10.50	-	-	-
T3-2nd-6	624	20.08	8.89	10.15	6.35	74.34	25.70
T3-2nd-4 PCA [11]	37	17.24	-	9.71	-	-	-
T3-2nd-6 PCA [11]	42	17.14	-	9.49	-	-	-
BiCE	4608 (B)	14.47	7.50	8.34	6.01	48.66	17.77
BiCE PCA	256	12.76	7.03	7.46	5.72	45.04	16.74
BiCE PCA	128	13.85	7.24	8.01	5.95	47.69	17.44
BiCE PCA	64	15.82	7.90	9.97	6.50	49.12	18.94
BiCE PCA	32	20.15	9.09	14.37	7.58	54.51	20.84

Gaussian interest point detectors [1] from the Statue of Liberty and Notre Dame cathedral, as shown in Figure 2(a,b). Pairs of matching image patches are verified using structure from motion [4]. These datasets are effective for measuring a patch descriptor's robustness to lighting variation and changes in viewpoint.

We created an additional dataset to measure robustness to intra-category appearance variation, as shown in Figure 2(c). The category dataset consists of 20 collections of 64×64 patches extracted from the Caltech 256 [30] dataset. Each collection of patches is selected by humans from a single category centered on the same part of the object, e.g. the back wheel of a motorcycle, the head of a turtle, etc. From these sets, 12,800 positive patch pairs are split to create testing and training datasets. An equal number of negative patch pairs are also generated using random patch selection. The dataset is available from the author's website.

Table 1 shows the results of various patch descriptors on the three datasets. We compare our approach Binary Coherent Edge descriptor (BiCE), to SIFT [1], Gloh [9], and several state-of-the-art descriptors T2-8a-2r6s, T3-2nd-4, T3-2nd-6 from [11]. Error rates at 95% recall and Equal Error Rates (EER) are given. The EER is the point on the ROC curve where the percentage of false positives and false negatives are equal. Since we are using 64×64 patches we also computed SIFT 64 and Gloh 64 using their standard resolutions for spatial binning, but with the full resolution patches for a fair comparison. Results of our descriptor using smaller patches and other variations are shown in the next section. Results using PCA with various dimensions are also shown. Parameters

are kept constant for all experiments using the values stated in previous sections. Running times for computing a descriptor were approximately 11ms for BiCE, 2ms for SIFT and 14ms for T3-2nd-6 on a 2.4GHz Intel PC. The code for BiCE is only partially optimized.

The best results are found across all datasets using BiCE with PCA and 256 dimensions, followed closely by BiCE without PCA. The results for T3-2nd-4 and T3-2nd-6 with PCA also perform well. However, rotating these descriptors using PCA in high dimensional spaces can be computationally expensive. It is worth noting that T2-8a-2r6s does relatively better on the category dataset than other previous methods. We hypothesize this is due to the inhibition technique used to compute orientation binning.

5.1 Parameter Exploration

In this section, we explore various adjustments and parameter changes to the previously described approach. The results are summarized in Table 2. The first set of figures shows the result of various sampling densities on the histogram H to get our final descriptor D. The results show that additional sampling in the y' dimension does not provide additional accuracy. As the sampling rate decreases the accuracies slowly decrease. Even with only 432 binary dimensions (54 bytes of storage) the accuracies still outperform previous techniques. The value of τ is varied from 10% to 30%, with only minor differences in accuracies. The removal of the edge length dimension increases the error rate by approximately 6% at 95% recall. The direct use of normalized continuous values sampled from H instead of using binarization significantly increases the 95% error rate to 27.42%. Similar to the binarization stage, the bins corresponding to different edge lengths were normalized independently. Normalizing all values together produces worse results. We also tried binarizing T3-2nd-6 [11] and SIFT [1] features using our simple approach, but improved results were not achieved. Other more sophisticated approaches to binarization could produce better results [23,24]. Finally we tested the descriptor's invariance to the initial patch size. As the patch size decreases, the accuracies are slightly better (32×32) or slightly worse (18×18).

5.2 Min-hash

The results using the min-hash approach from Section 4.2 are summarized in Table 3. ROC curves for a subset of the results can be seen in Figure 4, with BiCE providing an upper bound on the accuracies. The "% Match" and "% Non-match" columns indicate the probability of an descriptor having a corresponding hash value if it is a matching or non-matching descriptor. For instance, if a dataset had 1 million descriptors with most being non-matches, we would find on average 155,500 descriptors in each entry of the inverse lookup table using a sketch of size 1. As we can see, sketches of larger size are advantageous to minimize collisions. However, larger sketches also require more hashes to be stored to find collisions with correct matches. The right tradeoffs are application dependent. It is interesting to note that the min-hash approach produces

Table 2. Variations of parameters and methods on the BiCE baseline algorithm. This includes different descriptor sizes, differing values of τ, removal of edge length information, using continuous values instead of binary and using various patch sizes. (B) denotes binary dimensions.

		Liberty	
Method	Dimensions	95% Error rate	EER
BiCE baseline	4608 (B)	14.47	7.50
BiCE $n_{x'} = 24, n_{y'} = 24, n_\theta = 12, \sigma_{y'} = 1$	13824 (B)	14.68	7.67
BiCE $n_{x'} = 16, n_{y'} = 4, n_\theta = 8$	1024 (B)	15.22	7.72
BiCE $n_{x'} = 12, n_{y'} = 3, n_\theta = 6, \sigma_{x'} = 1.5, \sigma_{y'} = 4, \sigma_\theta = 1.5$	432 (B)	16.27	7.90
BiCE $\tau = 10\%$	4608 (B)	16.28	7.82
BiCE $\tau = 15\%$	4608 (B)	14.86	7.52
BiCE $\tau = 30\%$	4608 (B)	14.46	7.63
BiCE $n_l = 1$	2304 (B)	20.36	9.55
BiCE Continuous	4608	27.42	10.27
T3-2nd-6 Binary, $\tau = 20\%$	624 (B)	20.00	8.89
SIFT Binary, $\tau = 20\%$	128 (B)	39.65	13.88
BiCE 32×32 patch	4608 (B)	13.86	7.41
BiCE 18×18 patch	4608 (B)	16.03	7.84

Table 3. Error rates at 95% recall and Equal Error Rates (EER) for various sketch sizes and numbers of sketches on the Liberty dataset. The percentage of match and non-match image patches sharing a sketch on average.

		Liberty			
Sketch size	Number of sketches	95% Error rate	EER	% Match	% Non-match
1	32	42.03	19.26	40.47	15.55
1	64	33.96	11.13		
2	64	60.95	17.08	18.57	2.91
2	128	27.78	11.79		
3	128	44.36	19.67	9.23	0.65
3	256	47.94	12.14		
4	256	50.60	17.62	4.75	0.15
4	512	37.53	15.37		

similar accuracies to SIFT using 128 sketches of size 2 or 64 sketches of size 1. Hashing techniques are ideal for applications that can handle some degradation in matching accuracy for gains in efficiency, such as large scale image clustering and near-duplicate image search [26,27].

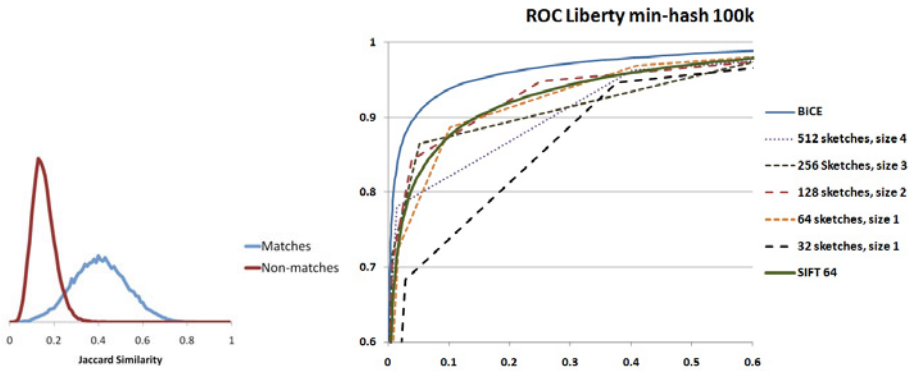


Fig. 4. (a) Density functions for matching and non-matching image pairs with respect to their Jaccard similarity, (b) ROC curves for various numbers of sketches and sizes. BiCE provides an upper bound on the accuracy of the min-hashing approaches.

6 Discussion and Conclusion

In this paper, we have developed a simple and effective image patch descriptor that provides state-of-the-art results. The descriptor encodes edge position, orientation, and local linear length, but not relative gradient magnitudes. We describe two techniques for dimensionality reduction using PCA and min-hash. Min-hash also provides a method for efficient patch retrieval.

In designing the descriptor, we experimented with other edge information such as curvature and distinguishing between even and odd edges. However, these approaches did not yield improved results. For category recognition, it can be important to be invariant to edge polarity, which our descriptor is not. It is still an open question on how to encode robustness in situations where it is useful while not providing full invariance when polarity is informative.

Finally, our edge descriptor might be invariant to relative gradient magnitudes, but interest point detectors are generally not with some exceptions [31]. An area of future work is to develop a corresponding interest point detector for sparse sampling that is robust to relative gradient magnitude and intensity changes.

References

1. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
2. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR, pp. 2161–2168 (2006)
3. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. IJCV 66 (2006)
4. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3d. ACM Transactions on Graphics 25, 835–846 (2006)
5. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR (2003)

6. Crandall, D., Huttenlocher, D.: Weakly supervised learning of part-based spatial models for visual object recognition. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 16–29. Springer, Heidelberg (2006)
7. Felzenszwalb, P., Mcallester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR (2008)
8. Berg, A.C., Malik, J.: Geometric blur for template matching. In: CVPR, pp. 607–614 (2001)
9. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE PAMI 27, 1615–1630 (2005)
10. Winder, S.A.J., Brown, M.: Learning local image descriptors. In: CVPR (2007)
11. Winder, S., Hua, G., Brown, M.: Picking the best daisy. In: CVPR, pp. 178–185 (2009)
12. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)
13. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE PAMI 27, 1615–1630 (2005)
14. Ke, Y., Sukthankar, R.: PCA-SIFT: A more distinctive representation for local image descriptors. In: CVPR, pp. 506–513 (2004)
15. Tola, E., Lepetit, V., Fua, P.: A fast local descriptor for dense matching. In: CVPR (2008)
16. Bay, H., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
17. Osindero, S., Hinton, G.E.: Modeling image patches with a directed hierarchy of markov random fields. In: NIPS 20 (2008)
18. Lepetit, V., Fua, P.: Keypoint recognition using randomized trees. IEEE PAMI 28, 1465–1479 (2006)
19. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: CVPR (2008)
20. Babenko, B., Dollar, P., Belongie, S.: Task specific local region matching. In: ICCV (2007)
21. Cormen, T.H.: Introduction to Algorithms. MIT Press, Cambridge (2001)
22. Torralba, A., Fergus, R., Weiss, Y.: Small codes and large databases for object recognition. In: CVPR (2008)
23. Salakhutdinov, R., Hinton, G.: Semantic hashing. Int. J. of Approximate Reasoning (2009)
24. Raginsky, M., Lazebnik, S.: Locality-sensitive binary codes from shift-invariant kernels. In: NIPS (2009)
25. Broder, A.Z.: On the resemblance and containment of documents. In: Compression and Complexity of Sequences (SEQUENCES 1997), pp. 21–29. IEEE Computer Society, Los Alamitos (1997)
26. Chum, O., Philbin, J., Zisserman, A.: Near duplicate image detection: min-hash and tf-idf weighting. In: British Machine Vision Conference (2008)
27. Chum, O., Perdoch, M., Matas, J.: Geometric min-hashing: Finding a (thick) needle in a haystack. In: CVPR, pp. 17–24 (2009)
28. Kulis, B., Grauman, K.: Kernelized locality-sensitive hashing for scalable image search. In: ICCV (2009)
29. Silpa-Anan, C., Hartley, R.: Optimised KD-trees for fast image descriptor matching. In: CVPR, pp. 1–8 (2008)
30. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology (2007)
31. Mikolajczyk, K., Zisserman, A., Schmid, C.: Shape recognition with edge-based features. In: British Machine Vision Conference (2003)

Adaptive and Generic Corner Detection Based on the Accelerated Segment Test

Elmar Mair[1,⋆], Gregory D. Hager[2], Darius Burschka[1],
Michael Suppa[3], and Gerhard Hirzinger[3]

[1] Technische Universität München (TUM), Department of Computer Science,
Boltzmannstr. 3, 85748 Garching bei München, Germany
{elmar.mair,burschka}@cs.tum.edu
[2] Johns Hopkins University (JHU), Department of Computer Science,
3400 N. Charles St., Baltimore, MD 21218-2686, USA
hager@cs.jhu.edu
[3] German Aerospace Center (DLR), Institute of Robotics and Mechatronics,
Münchner Str. 20, 82230 Wessling, Germany
{michael.suppa,gerd.hirzinger}@dlr.de

Abstract. The efficient detection of interesting features is a crucial step for various tasks in Computer Vision. Corners are favored cues due to their two dimensional constraint and fast algorithms to detect them. Recently, a novel corner detection approach, FAST, has been presented which outperforms previous algorithms in both computational performance and repeatability. We will show how the accelerated segment test, which underlies FAST, can be significantly improved by making it more generic while increasing its performance. We do so by finding the optimal decision tree in an extended configuration space, and demonstrating how specialized trees can be combined to yield an adaptive and generic accelerated segment test. The resulting method provides high performance for arbitrary environments and so unlike FAST does not have to be adapted to a specific scene structure. We will also discuss how different test patterns affect the corner response of the accelerated segment test.

Keywords: corner detector, AGAST, adaptive, generic, efficient, AST.

1 Introduction

Efficient corner detection algorithms are the basis for many Computer Vision applications, *e.g.* to find features for tracking, tracking by matching, augmented reality, registration or 3D reconstruction methods. Compared to edges and color cues, corners are more accurate and provide a two dimensional constraint. Considering corners as intersection of two edges, these features have no spatial extension and, therefore, there is no ambiguity in their location. Of course, this

⋆ This work was performed during the author's research stay at the CIRL lab (Johns Hopkins University). We are deeply grateful to all members of this research group for the interesting discussions and the great support.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 183–196, 2010.

aspect is only valid if the locality of a corner is preserved and the response of a corner detector is as close as possible to the real corner location. Several different approaches to corner detection are known in literature. All try to find a solution for efficient, accurate and reliable corner detection - three rather conflicting characteristics.

The Harris corner detection algorithm is probably one of the most popular corner extraction methods [1]. It is based on the first order Taylor expansion of the second derivative of the local sum of squared differences (SSD). The eigenvalues of this linear transformation reveal how much the SSD approximation varies if the patch would be shifted along the image axes. There are solutions which interpret the eigenvalues based on a threshold [2] or without [3]. So called global matching algorithms allow features to be detected within the whole image. Therefore, a corner detector has to provide a high repeatability so that it detects the same features also after large affine transformations. The global tracker SIFT [4] uses difference of Gaussians (DoG), while the faster SURF [5] uses a Haar wavelet approximation of the determinant of the Hessian. Both methods have the drawback of being rather computationally expensive. Smith developed the so called "Smallest Uni-Value Segment Assimilating Nucleus Test" (SUSAN) [6] for corner detection. The brightness of the center pixel, the nucleus, is compared to its circular pixel neighborhood, and the area of the uni-value segment assimilating nucleus (USAN) is computed. Corner and edges can be detected by evaluating this area, or it can also be used for noise reduction. The advantages of this approach are that no noise sensitive derivation or other computationally expensive operations have to be performed. In [6] a circular disc with diameter 3.4 is used, which yields a total area of 37 pixels. A more comprehensive survey can be found in [7].

In the last decade the processing power of standard computers has become fast enough to provide corner extraction at video rate. However, running conventional corner detection (*i.e.* the Harris corner detector) and performing other intensive tasks, is computationally infeasible on a single processor. With the introduction of recent techniques such as the "Features from Accelerated Segment Test" (FAST) [8], feature extraction has seen significant performance increase for real-time Computer Vision applications. While being efficient, this method has proven in several applications to be reliable due to high repeatability (see [9]). Some applications which use FAST are, *e.g.*, Klein's PTAM [10] and Taylor's robust feature matching in 2.3 μs [11].

In this work we are going to present a novel corner detection approach, which is based on the same corner criterion as FAST, but which provides a significantly *performance increase* for *arbitrary* images. Unlike FAST, the corner detector does not have to be trained for a specific scene, but it *dynamically adapts* to the environment while processing an image.

Section 2 discusses FAST in more detail due to its strong relation to the presented work. In Section 3, we will present the *adaptive* and *generic* accelerated segment test with increased performance for arbitrary environments. Further,

we will discuss the use of different segment pattern and show some experimental results to demonstrate the achieved speed-up in Section 4.

2 FAST Revisited

The FAST principle is based on the SUSAN corner detector. Again, the center of a circular area is used to determine brighter and darker neighboring pixels. However, in the case of FAST, not the whole area of the circle is evaluated, but only the pixels on the discretized circle describing the segment. Like SUSAN, also FAST uses a Bresenham's circle of diameter 3.4 pixels as test mask. Thus, for a full accelerated segment test 16 pixels have to be compared to the value of the nucleus. To prevent this extensive test, the corner criterion has been even more relaxed. The criteria for a pixel to be a corner according to the accelerated segment test (AST) is as follows: there must be at least S connected pixels on the circle which are brighter or darker than a threshold determined by the center pixel value. The values of the other $16 - S$ pixels are disregarded. Therefore, the value S defines the maximum angle of the detected corner. Keeping S as large as possible, while still suppressing edges (where $S = 8$), increases the repeatability of the corner detector. Thus, FAST with segment size 9 (FAST-9) is usually the preferred version, and is also used in our experiments unless otherwise stated. The AST applies a minimum difference threshold (t) when comparing the value of a pixel on the circular pattern with the brightness of the nucleus. This parameter controls the sensitivity of the corner response. A large t-value results in few but therefore only strong corners, while a small t-value yields also corners with smoother gradients. In [9] is shown, that the AST with $S = 9$ has a high repeatability, compared to other corner detectors as, *e.g.*, Harris, DoG, or SUSAN. The repeatability of a corner detector is a quality criterion which measures the capability of a method to detect the same corners of a scene from varying viewpoints.

One question still remains, namely which pixel to compare first, second, third, and so forth. Obviously, there is a difference in speed, whether one consecutive pixel after another is evaluated or, *e.g.*, bisection on the circle pattern is used to test if the corner criterion applies or cannot apply anymore. This kind of problem is known as constrained twenty questions paradigm. When to ask which question results in a decision tree with the aim to reduce its average path length. In [12], Rosten uses ID3 [13], a machine learning method, to find the best tree based on training data of the environment where FAST is applied. Doing so, it is not guaranteed that all possible pixel configurations are found (see Section 4.2). Already small rotations of the camera may yield pixel configurations which have not been measured in the test images. And even if all the pixel configurations are present, a small rotation about the optical axis would cause the probability distribution of the measured pixel configurations to change drastically. This may result in an incorrect and slow corner response. To learn the probabilistic distribution of a certain scene is therefore not applicable unless only the same viewpoints and the same scene are expected. Note that the decision tree is optimized for a specific

environment and has to be re-trained every time it changes to provide the best performance.

The decision tree learning used by the FAST algorithm builds a *ternary* tree with possible pixel states "darker", "brighter" and "similar". At each learning step, *both* questions, "is brighter" and "is darker", are applied for all remaining pixel and the one with the maximum information gain is chosen. Hence, the state of each pixel can be one of four possibilities: unknown (u), darker (d), brighter (b) or similar (s). In the following we call a combination of N such states a pixel *configuration*. The size of the configuration space is therefore 4^N, which yields $4^{16} \approx 4 \cdot 10^9$ possible configurations for $N = 16$. For the rest of this paper we refer to this model as restricted or four states configuration space.

FAST-ER, the most recent FAST derivation, has even a slightly increased repeatability, compared to FAST-9, at the cost of computational performance [9]. The main difference is the thickness of the Bresenham's circle, that has been increased to 3 pixels. This results again in a more SUSAN-like algorithm, which spans a circular area of 56 pixels, disregarding the inner 3x3 pixels. Again, ID3 is used to build the decision tree, restricting the evaluation to only a small part of the 47 pixels.

3 Adaptive and Generic Accelerated Segment Test

In this section we present a corner detection approach which is also based on the AST, but which is more efficient, while being more generic too. We introduce the reader step-wise to the different concepts underlying the algorithm.

3.1 Configuration Space for a Binary Search Tree

Instead of only considering a restricted configuration space, as in FAST, we propose to use a more detailed configuration space in order to provide a more efficient solution. To do this, we consider to evaluate a single question per time. The idea is as follows: choose one of the pixels to test and *one* question to pose. The question is then evaluated for this given pixel, and the response is used to decide the following pixel and question to query. Searching for a corner, hence, reduces to traversing a *binary* decision tree. Since, it is required to specify which pixel to query and the type of question to use. Consequently, the configuration space increases by the addition of two more states: "not brighter" ($\overline{b}$) and "not darker" ($\overline{d}$). Using a similar notion as [12], the state of a pixel relative to the nucleus n, denoted by $n \to x$, is assigned as follows:

$$S_{n\to x} = \begin{cases} d, & I_{n\to x} < I_n - t & \text{(darker)} \\ \overline{d}, & I_{n\to x} \not< I_n - t \bigwedge S'_{n\to x} = u & \text{(not darker)} \\ s, & I_{n\to x} \not< I_n - t \bigwedge S'_{n\to x} = \overline{b} & \text{(similar)} \\ s, & I_{n\to x} \not> I_n + t \bigwedge S'_{n\to x} = \overline{d} & \text{(similar)} \\ \overline{b}, & I_{n\to x} \not> I_n + t \bigwedge S'_{n\to x} = u & \text{(not brighter)} \\ b, & I_{n\to x} > I_n + t & \text{(brighter)} \end{cases} \tag{1}$$

where $S'_{n\to x}$ is the preceding state, I is the brightness of a pixel and u means that the state is still unknown. This results in a binary tree representation, as opposed to a ternary tree, allowing a single evaluation at each node. Note that this increases the configuration space size to 6^N, which yields $6^{16} \approx 2 \cdot 10^{12}$ possible nodes for $N = 16$.

Associated with each branch of our tree is a processing cost, which represents the computational cost on the target machine. These costs vary due to different memory access times. We specify these as follows,

- c_R: register access cost (second comparison of the last tested pixel),
- c_C: cache access cost (testing of a pixel in the same row)
- c_M: memory access cost (testing of any other pixel).

Further, for each of these, an additional cost equivalent to evaluating a greater-than operation, is required.

3.2 Building the Optimal Decision Tree

It is well known that a greedy algorithm, such as ID3, performs rather poorly when finding the optimal decision tree [14]. However, the issue of finding such a tree is a well-studied problem, where it has been shown that finding the global optimum is NP-complete [15]. There are several solutions towards finding the optimal tree [16,17,18], but they are either approximations to the global optimum or are restricted to special cases, making them ill-suited for this application.

In order to find the optimal decision tree we implemented an algorithm which is similar to the backward induction method [16]. We explore the whole configuration space starting at the root of the decision tree, where none of the pixels is known. Nodes of the tree are formed by recursively evaluating a possible question at a given pixel. We explore the configuration space (using Depth First Search) until a leaf is found, where a leaf is defined as the first node on the path which fulfills or cannot fulfill anymore the AST corner criteria. The cost at a given leaf is zero, while the cost at any given internal node, c_P, is determined by picking the minimum cost computed for each child pair C_+ and C_-, representing the positive and negative results of a test, by

$$c_P = \min_{\{(C_+,C_-)\}} c_{C_+} + p_{C_+} c_T + c_{C_-} + p_{C_-} c_T = c_{C_+} + c_{C_-} + p_P c_T \qquad (2)$$

where c_T represents the cost of the pixel evaluation with $c_T \in \{c_R, c_C, c_M\}$ and the p_P, p_{C_+} and p_{C_-} are the probabilities of the pixel configurations at the parent and child nodes respectively. Using this dynamic programming technique allows us to find the decision tree for an optimal AST (OAST) efficiently. The resulting decision tree can therefore be optimized for different c_R, c_C and c_M, but also for arbitrary probabilities for each pixel configuration, which is necessary for our approach described in the following section.

The binary configuration space allows for decision trees which reduce the entropy more quickly than a ternary tree, as questions which contain little information gain are deferred to later stages of the decision process. Note that the

additional cost of re-evaluating the same pixel at a subsequent point in time is taken into account when computing the optimal tree.

3.3 Adaptive Tree Switching

Every image has, independent of the scene, homogeneous and (or) cluttered areas representing uniform surfaces or structured regions with texture. Hence, instead of learning the distribution of the pixel configurations from training images, like FAST, a first generalization would be to learn the probability of structured and homogeneous regions and optimize the decision tree according to this distribution. The resulting tree is complete and optimized for the trained scene, while being invariant to camera rotations. The probability of an image to be uniform can be modeled by the probability of a pixel state to be similar to the nucleus (p_s). The "brighter" and "darker" states are mirrored states, which means that, *e.g.*, a brighter pixel on the test pattern will evaluate the current nucleus pixel as darker as soon as it becomes the center pixel. Due to this mirroring the states "brighter" and "darker" are assumed to have the same probability (p_{bd}), which is chosen to sum up to one with p_s ($p_s + 2p_{bd} = 1$). Thus, the probability of a pixel configuration p_X can be computed as follows:

$$p_X = \prod_{i=1}^{N} p_i \quad \text{with } p_i = \begin{cases} 1 & \text{for } S_{n\to i} = u \\ p_s & \text{for } S_{n\to i} = s \\ p_{bd} & \text{for } S_{n\to i} = d \bigvee S_{n\to i} = b \\ p_{bd} + p_s & \text{for } S_{n\to i} = \overline{d} \bigvee S_{n\to i} = \overline{b} \end{cases} \tag{3}$$

The probability distribution of the pixel configuration is therefore a trinomial distribution with the probabilities p_s and twice p_{bd}. Note that the states $\overline{d}$, $\overline{b}$ and u are not samples of this distribution but represent a set of two and three samples respectively. While this approach provides a good solution for the trained environment, it is not generic and, as FAST, it has to be learned for each specific scene where it is applied.

A more efficient and generic solution is achieved, if the algorithm automatically adapts to the area which is currently processed, *i.e.* it switches between decision trees which are optimized for the specific area. The idea is to build, *e.g.*, two trees and specialize one for homogeneous and one for structured regions based on a small and a large value for p_s. At the end of each decision path, where the corner criterion is met or cannot be fulfilled anymore, a jump to the appropriate specialized tree is performed based on the pixel configuration of this leaf (see Fig. 1). This switch between the specialized decision trees comes with *no* additional costs, because the evaluation of the leaf node is done offline when generating the specialized tree. In this way the AST is adapted to each image section dynamically and its performance is *increased*, for an *arbitrary* scene. Any learning becomes needless.

Because a switch between the trees at no costs can only be performed at a leaf, the adaption is delayed by one test. Therefore, the only case were the adaptive and generic accelerated segment test (AGAST) would be less efficient

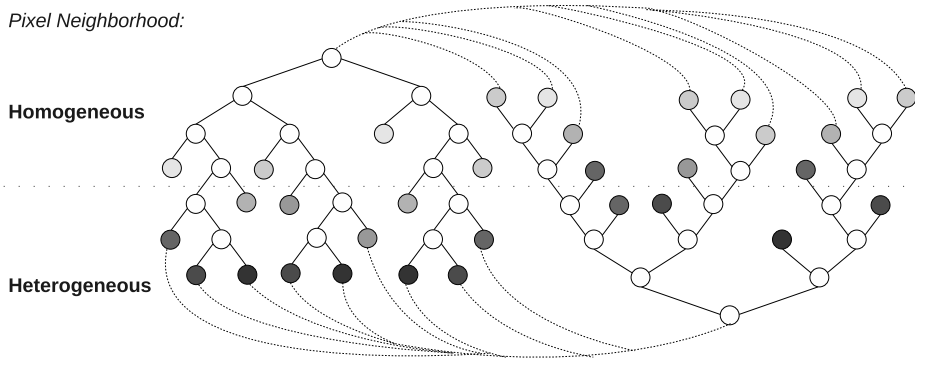


Fig. 1. Principle of the adaptive and generic accelerated segment test. The AGAST switches between two (or more) specialized trees as soon as the pixel neighborhood changes. The lighter the gray of a leaf the more equal pixels are in the configuration. The left tree achieves less pixel evaluations (shorter decision paths) in a homogeneous pixel neighborhood, while the right one is optimized for textured regions.

than FAST, is if the environment would switch from homogeneous to structured and vice versa at consecutive pixels. This is practically not possible, due to the mirroring effect of dissimilar pixels as described earlier. However, natural images usually do not have a random brightness distribution, but they are rather split into cluttered and uniform regions. If the decision trees can be strongly balanced by varying p_s, also more than two different weighted trees can be used.

4 Experimental Results

The speed and the repeatability of FAST have already been compared to state of the art corner detection algorithms in [9]. In those experiments FAST-9 has demonstrated better performance than, *e.g.*, Harris, DoG, or SUSAN. Thus, we renounce to compare our AST variation only with FAST-9. Note that our approach is also based on the AST and, therefore, it provides the same repeatability as FAST.

First, we show and discuss an experiment where we compare the performance of different AST masks on noisy and blurry images. In Section 4.2 we evaluate the corner response of different balanced decision trees; and, finally, we compare the performance of FAST with our approach.

4.1 Evaluation of Various AST Patterns

As already mentioned, SUSAN as well as FAST use a circle radius of 3.4 pixels. In [6] it is noted that the mask size does not influence the feature detection as long as there is no more than one feature within the mask. The effect of the mask size of an image operator is well studied for filters with dense masks.

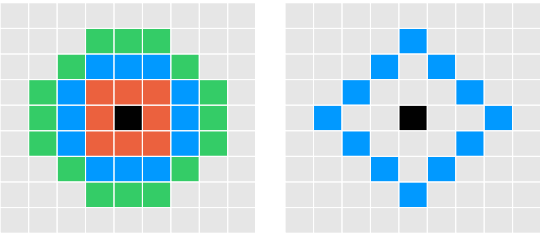


Fig. 2. Different mask sizes for the AST: a 4 pixels mask (red), a squared and diamond shaped 12 pixels mask (blue, left and right figure) and a 16 pixels mask (green). The black pixel represents the nucleus.

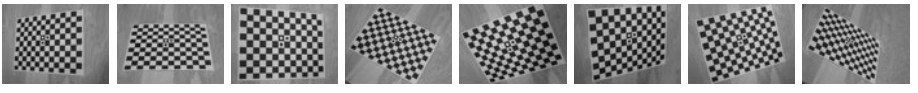
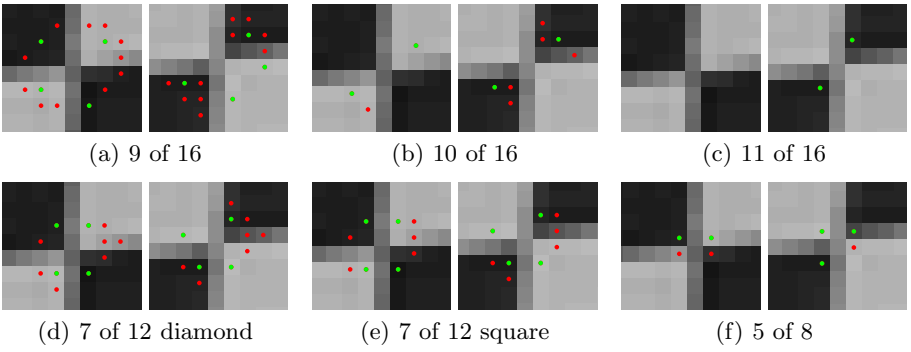


Fig. 3. Checkerboard dataset

Their size affects the smoothing behavior so that larger filters are more robust against noise. The corresponding effect for the AST pattern size has so far not been discussed in the similar literature. While for the dense mask of SUSAN, the same smoothing criteria as mentioned above apply, it is not obvious that large circles have a similar smoothing effect for AST. Therefore, we use eight checkerboard pictures acquired from different viewpoints (see Fig. 3) to evaluate the corner response of the AST pattern shown in Fig. 2. A checkerboard provides many bright and dark corners of different sizes if viewed from different angles. Further, we add Gaussian blur and noise to determine the performance of these pattern on images of poor quality. For all the tests the same threshold is applied.

For pattern sizes up to 12 pixels it is possible to compute the optimal path by exploring the six state configuration space as described in Section 3.1. The computational resources of conventional computers are not sufficient to find the optimal tree for a 16 pixel pattern within the extended configuration space in reasonable time. Thus, for this size we compute the optimal tree based on the four state space, yielding a ternary decision tree. Before generating the machine code, the tree is splatted as described in [9] to cut off equal branches.

Fig. 4 shows the corner response of a 16 pixel pattern with arc lengths of 9, 10 and 11 (12 is omitted because it does not find any features at these corners), the 12 pixel pattern with a square and diamond shape as well as the 8 pixel pattern. The larger the mask and the larger the arc threshold S, the more features that are found. A small arc is more discriminating and yields features only close to the real corner location, which is apparent in Fig. 4(c). Large patterns result in multiple responses around a corner location, but they may lie at a distance of about the radius of the mask from the real corner (see Fig. 4(a)). Thus, they do not preserve the corner location. They are therefore slower for two reasons: 1) the



(a) 9 of 16 (b) 10 of 16 (c) 11 of 16

(d) 7 of 12 diamond (e) 7 of 12 square (f) 5 of 8

Fig. 4. The corner response for different AST pattern. Detected features are colored in red. The corners after non-maximum suppression are green.

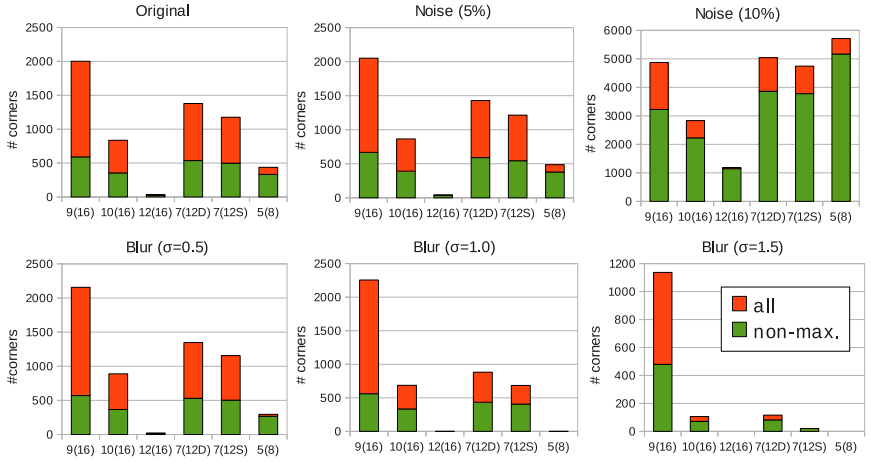


Fig. 5. These charts compare the corner response of different patterns for blurry and noisy images. To preserve the comparability we use the arc length S and in brackets the mask size to label the bars. The red bars show the total amount of features found, while the green bars represent the number of corners after non-maximum suppression. Note that the scales of the charts are not the same.

processing of a large pattern is of course computationally more expensive, and 2) they need to evaluate many features for non-maximum suppression. Smaller patterns better preserve the locality constraint of a corner. However, in the case of the pattern of size 8, the features are too close, so that a part of them get lost after non-maximum suppression. Thus, for this size such a post processing is not necessary, because only single responses are observed at a corner, and should even be avoided to prevent the loss of features.

For the next experiment the original checkerboard images were modified by adding Gaussian noise (5% and 10%) and Gaussian blur ($\sigma = 0.5, 1.0, 1.5$). Fig. 5 shows the performance of the different pattern on these images. Here the advantage of the 16 pixel mask with arc length 9, 9(16), becomes apparent. It is more robust against noise and blur. However, the same mask sizes but with larger arcs show a similar drop-off on blurry images as the smaller pattern with similar segment angle. The 16 pixel mask with arc length 12, 12(16), has a similar arc angle as 5(8), while 7(12) has a similar angle as 16(10).

The size of the arc angle controls the repeatability, as shown in [9], and the robustness against blur. The arc length and, thus, the radius of the mask influences the robustness against noise.

4.2 Corner Response Time

The corner response time of a certain decision tree is evaluated by computing the number of tests (greater-than or less-than evaluations) for all the possible pixel configurations of a mask. To compare the weighting effects of different probabilities of a pixel to be similar (p_s), as described in Section 3.3, the pixel configurations are divided into classes representing the number of similar pixels. Fig. 6 shows the deviation of the mean and the standard deviation of the corner response time from the minimum of all tests on a class. The trees are built for 12 pixel masks exploring the six state configuration space. For zero or one similar pixels the trees with weight $p_s = 0.1$ and $p_s = 0.01$ perform fewer tests as trees with larger values for p_s. Also the standard deviation of the classes is smaller for these trees. It is apparent that the decision trees can not be balanced significantly due to the strong symmetry of this special constrained twenty questions problem. Besides, the classes with a large number of similar pixels cannot be balanced properly anymore, because the amount of possible configurations decreases drastically for them. Nevertheless, the performance of the adaptive tree is better than if only one tree is used, as we will see in Section 4.3.

No performance increase can be achieved for different p_s by exploring only the restricted configuration space, due to the reduced degrees of freedom compared to the full six state configuration space. The limitations of the latter space are also apparent in the performance of the tree M12 (4st), which shows a significantly higher average of tests as M12 (6st) in Table 1. This table compares the corner response time for trees of various mask sizes which were built using different methods. The second data row M12 (6st) shows the minimum time of the trees compared in Fig. 6 which were specialized for different p_s. Thus, these values are achieved using the AGAST, switching between two trees which are optimized for $p_s = 0.1$ and $p_s = 1/3$.

Experiments have shown, that by learning a decision tree based on 120 outdoor images as proposed in [12], only about 87000 pixel configurations out of over 43 million possible ones could be found. Any learned decision tree should therefore be enhanced by the missing configurations to prevent false positive and false negative responses. The ID3 based decision tree, learned from all possible configurations with equal weights, has shown to achieve the best corner response

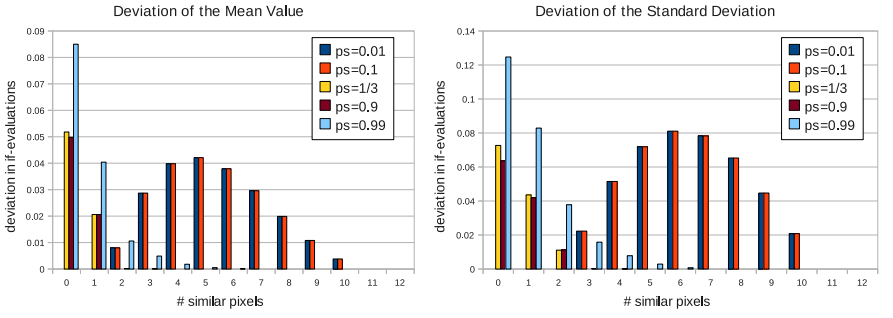


Fig. 6. This chart illustrates the performance of various decision trees, based on different probabilities for a pixel to be similar (p_s). Each decision tree was tested with all possible pixel combination for the mask.

(a) Lab (b) Outdoor (c) Indoor (d) Aerial (e) Medical

Fig. 7. The scenes used for the performance test are a lab scene (768x288), an outdoor image (640x480), an indoor environment (780x580), an aerial photo (780x582) and an image from a medical application (370x370)

of all trees witch were optimized using ID3 and various p_s. Indeed, it yields the identical corner response as the code provided in the FAST sources.[1]

4.3 Performance Experiments

All the timing experiments are run on one core of an Intel Core2 Duo (P8600) processor at $2.40\,GHz$. We are using five images from different scenes[2], shown in Fig. 7.

Table 2 shows the performance of various AST-decision trees with different mask sizes and built by different methods. Please note, that the achieved speed-ups do not only affect the corner detection step, but also the computation of the pixel-score for the non-maximum suppression.

To compare the performance of our decision trees with the conventional FAST-9 algorithm, we use the code from the FAST sources mentioned in Section 4.2. The FAST and optimal AST (OAST) trees are built based on a uniform probability distribution, which means that the probability for any pixel configuration

[1] http://svr-www.eng.cam.ac.uk/~er258/work/fast.html
[2] The lab scene is provided in the FAST Matlab package at http://svr-www.eng.cam.ac.uk/~er258/work/fast-matlab-src-2.0.zip

Table 1. This table compares the average tests performed for each class of configuration using various mask sizes and different methods to find the best decision tree. From left to the right: mask size 8 exploring the six states configuration space, mask size 12 exploring the six states space, mask size 12 exploring the four states space, mask size 16 exploring the four states space and the ID3-learned decision tree trained on all possible configurations. The probability of all configurations was assumed to be equal for all trees beside M12 (6st), which represents the minimum tests for all decision trees of Fig. 6. These trees were built by exploring the six state configuration space for a mask size of 12 pixels using different weights.

n_s	M8 (6st)	M12 (6st)	M12 (4st)	M16 (4st)	M16 (ID3)
0	5.54	6.53	7.80	8.1528	8.3651
1	5.32	6.17	7.27	7.6485	7.8073
2	5.07	5.82	6.77	7.1948	7.3094
3	4.81	5.48	6.33	6.7893	6.8692
4	4.59	5.19	5.93	6.4277	6.4812
5	4.41	4.94	5.59	6.1044	6.1388
6	4.26	4.74	5.28	5.8144	5.8354
7	4.13	4.56	5.01	5.5529	5.5649
8	4.00	4.41	4.77	5.3160	5.3223
9	-	4.29	4.55	5.1003	5.1033
10	-	4.18	4.35	4.9031	4.9043
11	-	4.08	4.17	4.7221	4.7225
12	-	4.00	4.00	4.5554	4.5555
13	-	-	-	4.4013	4.4013
14	-	-	-	4.2583	4.2583
15	-	-	-	4.1250	4.1250
16	-	-	-	4.0000	4.0000

is the same. This probability distribution yielded the trees with the best overall corner response and therefore the best performance.

As mentioned earlier, it is not possible to search for the optimal decision tree for a 16 pixel mask within the complete configuration space in reasonable time on conventional computer. Therefore, the tree is optimized in the four state configuration space and achieves an average speed-up of about 13% regarding FAST-9. For the 12 pixels mask the ideal tree can be found in the six state space and by combining the trees specialized for $p_s = 1/3$ and $p_s = 0.1$ a mean speed-up of about 23% and up to more than 30% can be gained. Using the AGAST-5 decision tree on the 8 pixels mask results in a performance increase of up to almost 50%. Of course, with the drawback of its sensitivity regarding noise and blur as discussed in Section 4.1.

The C-sources for OAST-9, AGAST-7 and AGAST-5 are available for download at `http://www6.cs.tum.edu/Main/ResearchAgast`. The trees have been optimized according to standard ratios of memory access times.

Table 2. This table shows the computational time of various AST-decision trees. The value in parentheses, close to the tree names, stands for the mask size which the tree is based on. The specified speedup is relative to the FAST performance. The first value represents the mean speedup for all five images while the value in parentheses shows the maximum speedup measured.

Image	Lab	Outdoor	Indoor	Aerial	Medical	Speed-Up [%]
FAST-9 (16)	1.8867	2.4242	1.8516	2.2798	1.1106	-
OAST-9 (16)	1.5384	2.2970	1.6197	1.9225	0.9413	13.4 (18.5)
AGAST-7 (12)	1.2686	1.9416	1.4405	1.8865	0.8574	23.0 (32.8)
AGAST-5 (8)	0.9670	1.4582	1.3330	1.8742	0.7727	33.0 (48.7)

5 Conclusion and Future Work

We have shown how to increase the performance of the accelerated segment test by combining specialized decision trees. The optimal trees are found by exploiting the full binary configuration space. The algorithm dynamically adapts to an arbitrary scene which makes the accelerated segment test generic. In doing so no additional costs arise. This makes this approach to the currently most efficient corner detection algorithm to our knowledge. Moreover, any decision tree learning to adapt to an environment becomes needless. By exploring the full configuration space also the processor architecture and its memory access times can be taken into account to yield the best performance on a specific target machine.

Further, we have discussed the influence of different AST mask sizes and shown that, for images of good quality, smaller mask sizes should be preferred. They reduce the processing time and emphasize the locality constraint of a corner. Dealing with blurry and noisy images, patterns with a larger radius are favored.

For future research we would like to implement an approximation for decision tree learning as proposed in [17], which considers also the length of the decision path and not only the minimization of the entropy, as ID3. In this way, we can also balance trees of pattern sizes 16 or more pixels and implement the AGAST for these masks. Further, we are looking for an efficient combination of different mask sizes to yield high robustness while preserving the real corner location.

Acknowledgments

We want to particularly acknowledge Raphael Sznitman for the fruitful discussions and his support. Further, we want to thank Frank Sehnke and the TUM-Cogbotlab group for allowing us to use one of their PCs, which sped up the development significantly.

This work was support by the DLR internal funding for image-based navigation systems.

References

1. Harris, C., Stephens, M.: A combined corner and edge detection. In: Proceedings of The Fourth Alvey Vision Conference, pp. 147–151 (1988)
2. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1994), pp. 593–600 (1994)
3. Noble, A.: Descriptions of Image Surfaces. PhD thesis, Department of Engineering Science, Oxford University (1989)
4. Lowe, D.G.: Object recognition from local scale-invariant features. International Journal of Computer Vision 60, 91–110 (2004)
5. Bay, H., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
6. Smith, S.M., Brady, J.M.: Susan - a new approach to low level image processing. International Journal of Computer Vision 23, 45–78 (1997)
7. Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: a survey. Foundations and Trends in Computer Graphics and Vision 3, 177–280 (2008)
8. Rosten, E., Drummond, T.: Fusing points and lines for high performance tracking. In: IEEE International Conference on Computer Vision (ICCV 2005), vol. 2, pp. 1508–1511 (2005)
9. Rosten, E., Porter, R., Drummond, T.: Faster and better: A machine learning approach to corner detection. IEEE Trans. Pattern Analysis and Machine Intelligence, PAMI (2009)
10. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2007), Nara, Japan (2007)
11. Taylor, S., Rosten, E., Drummond, T.: Robust feature matching in 2.3μs. In: IEEE CVPR Workshop on Feature Detectors and Descriptors: The State Of The Art and Beyond (2009)
12. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 430–443. Springer, Heidelberg (2006)
13. Quinlan, J.R.: Induction of decision trees. Machine Learning (1986)
14. Garey, M.R., Graham, R.L.: Performance bounds on the splitting algorithm for binary testing. Acta Informatica 3, 347–355 (1974)
15. Hyafil, L., Rivest, R.L.: Constructing optimal binary decision trees is np-complete. Information Processing Letters 5, 15–17 (1976)
16. Garey, M.R.: Optimal binary identification procedures. SIAM Journal on Applied Mathematics 23, 173–186 (1972)
17. Geman, D., Jedynak, B.: Model-based classification trees. IEEE Transactions on Information Theory 47 (2001)
18. Kislitsyn, S.S.: On discrete search problems. Cybernetics and Systems Analysis 2, 52–57 (1966)

Spatially-Sensitive Affine-Invariant Image Descriptors

Alexander M. Bronstein[1,2] and Michael M. Bronstein[1,3]

[1] BBK Technologies ltd.
[2] Dept. of Electrical Engineering, Tel Aviv University
[3] Dept. of Computer Science, Technion – Israel Institute of Technology

Abstract. Invariant image descriptors play an important role in many computer vision and pattern recognition problems such as image search and retrieval. A dominant paradigm today is that of "bags of features", a representation of images as distributions of primitive visual elements. The main disadvantage of this approach is the loss of spatial relations between features, which often carry important information about the image. In this paper, we show how to construct *spatially-sensitive* image descriptors in which both the features and their relation are affine-invariant. Our construction is based on a vocabulary of pairs of features coupled with a vocabulary of invariant spatial relations between the features. Experimental results show the advantage of our approach in image retrieval applications.

1 Introduction

Recent works [1,2,3,4,5,6,7,8,9] demonstrated that images can be efficiently represented and compared using local features, capturing the most distinctive and dominant structures in the image. The construction of a feature-based representation of an image typically consists of *feature detection* and *feature description*, often combined into a single algorithm. The main goal of a feature detector is to find stable points or regions in an image that carry significant information on one hand and can be repeatedly found under transformations. Transformations typically considered include scale [3,4], rotation, and affine [7,8] transformations. A feature descriptor is constructed using local image information in the neighborhood of the feature points (or regions).

One of the advantages of feature-based representations is that they allow to think of images as a collection of primitive elements (*visual words*), and hence appeal to the analogy of text search and use well-developed methods from that community. Images can be represented as a collection of visual words indexed in a "visual vocabulary" by vector quantization in the descriptor space [10,11]. Counting the frequency of the visual word occurrence in the image, a representation referred to as a *bag of features* (analogous to a *bag of words* used in search engines) is constructed. Images containing similar visual information tend to have similar features, and thus comparing bags of features allows to retrieve similar images.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 197–208, 2010.

Using invariant feature detectors and descriptors, invariance is built into bags of features by construction. For example, given two images differing by an affine transformation, their bag of features representations based on MSER descriptors are (at least theoretically) equal. Yet, one of the main disadvantages of bags of features is the fact that they consider only the statistics of visual words and lose the spatial relations between them. This may often result in loss of discriminativity, as spatial configuration of features often carries important information about the underlying image [12]. A similar problem is also encountered in text search problems. For example, in a document about "matrix decomposition" the word "matrix" is frequent. Yet, a document about the movie *Matrix* will also contain this word, which will result in a similar word statistics and, consequently, similar bags of features. In the most pathological case, a random permutation of words in a text will produce identical bags of words. In order to overcome this problem, text search engines commonly use vocabularies consisting not only of single words but also of combinations of words or *expressions*.

This text analogy can be extended to images. Unlike text which is one-dimensional, *visual expressions* are more complicated since the spatial relations of objects in images are two-dimensional. A few recent papers tried to extend bags of features taking into consideration spatial information about the features. Marszalek and Schmid [13] used spatial weighting to reduce the influence of background clutter (a similar approach was proposed in [14]). Grauman and Darrell [15] proposed comparing distributions of local features using *earth mover's distance* (EMD) [16], which incorporates spatial distances. Nister and Stewenius [17] used feature grouping to increase the discriminativity of image descriptors, and also showed that such the advantage of such an approach over enlarging the descriptor area is smaller sensitivity to occlusion. A similar approach for feature grouping and geometry consistency verification has been more recently proposed by Wu *et al.* [18]. Sivic *et al.* [19,20] used feature configurations for object retrieval. Chum and Matas [21] considered a special case when the feature appearance is ignored and only geometry of feature pairs is considered. In [22], the spatial structure of features was captured using a multiscale bag of features construction. The representation proposed in [23] used spatial relations between parts. In [24], in a different application of 3D shape description, spatially-sensitive bags of features based on pairs of words were introduced. Behmo *et al.* [25] proposed a *commute graph* representation partially preserving the spatial information. However, the commute graph based on Euclidean distance relations is not invariant under affine transformations. Moreover, commute graphs encode only translational relations between features, ignoring more complicated relations such as scale and orientation of one feature with respect to another.

The main focus of this paper is the construction of affine-invariant feature-based image descriptors that incorporate spatial relations between features. Our construction is based on a vocabulary of pairs of features coupled with a vocabulary of affine-invariant spatial relations. Such a construction is a meta-approach which can augment existing feature description methods and can be considered as an extension of the classical bags of features. The rest of the paper is organized

as follows. In Section 2, we introduce notation and the notions of invariance and covariance, using which we formally define feature detection, description, and bags of features. Section 3 describes our construction of affine-invariant spatially-sensitive bags of features. Section 4 demonstrates the performance of our approach in an invariant image retrieval experiment. Finally, Section 5 concludes the paper.

2 Background

Typically, in the computation of a bag of features representation of an image, first a *feature detector* finds stable regions in the image. Next, each of the detected features undergoes is transformed to an invariant *canonical representation*, from which a *visual descriptor* is computed. Each such descriptor containing visual information about the feature is quantized in a *visual vocabulary*, increasing the count of the visual word corresponding to it. Finally, counts from all features are collected into a single distribution, called a *bag of features*. In what follows, we formalize each of these steps.

Feature detection. Let us be given an image I (for simplicity, grayscale). We refer to a planar subset F as to a *feature*, and denote by $\mathbf{F}_I = \{F_1, \ldots, F_n\}$ a *feature transform* of I that produces a collection of features out of an image. The feature transform is said to be *covariant* with a certain group of geometric transformations if it commutes with action of the group, i.e., for every transformation $\mathbf{T}$, $\mathbf{F}_{\mathbf{T}I} = \mathbf{T}\mathbf{F}_I$ (we write $\mathbf{T}I(x)$ implying $I(\mathbf{T}x)$). In particular, we are interested in the group of affine transformations of the plane. We will henceforth assume that the feature transform is affine-covariant. A popular example of such a feature transform is MSER [7], which will be adopted in this study.

Feature canonization. Once features are detected, they are often normalized or *canonized* by means of a transformation into some common system of coordinates [26]. We denote the inverse of such a canonizing transformation associated with a feature F by $\mathbf{A}_F$, and refer to $\mathbf{A}_F^{-1}F$ as to a *canonical representation* of the feature. As before, this process is said to be affine-covariant if it commutes with the action of the affine group. The canonical representation in that case is *affine-invariant*, i.e., $\mathbf{A}_F^{-1}F = \mathbf{A}_{\mathbf{T}F}^{-1}(\mathbf{T}F)$ for every affine transformation $\mathbf{T}$. A classical affine-covariant (up to reflection ambiguity) feature canonization is based on zeroing its first-order moments (centroid) and diagonalizing the second-order moments [27].

Feature descriptors. The fact that a canonical representation of a feature is invariant is frequently used to create invariant descriptors. We will denote by $\mathbf{v}_F$ a vector representing the visual properties of the image supported on F and transformed by $\mathbf{A}_F^{-1}$ into the canonical system of coordinates, referring to it as to a *visual descriptor* of F. A straightforward descriptor can be obtained by simply sampling the feature footprint in the canonical space and representing the obtained samples in a vector form [26]. However, because of using the intensity values of the image directly, such a descriptor is sensitive to changes

in illumination. While this is not an issue in some applications, many real applications require more sophisticated representations. For example, the SIFT descriptor [3] computes a histogram of local oriented gradients (8 orientation bins for each of the 4×4 location bins) around the interest point, resulting in a 128-dimensional vector. SURF [9] descriptor is similar to SIFT yet more compact, with 4-dimensional representation for each of the 4×4 spatial locations (total of 64 dimensions).

Bags of features. Given an image, descriptors of its features are aggregated into a single statistic that describes the entire image. For that purpose, descriptors are vector-quantized in a *visual vocabulary* $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ containing m representative descriptors, which are usually found using clustering algorithms. We denote by $\mathbf{Q}_\mathbf{V}$ a quantization operator associated with the visual vocabulary $\mathbf{V}$ that maps a descriptor into a distribution over $\mathbf{V}$, represented as an m-dimensional vector. The simplest hard quantization is given by

$$(\mathbf{Q}_\mathbf{V}\mathbf{v})_i = \begin{cases} 1 : d(\mathbf{v}, \mathbf{v}_i) \leq d(\mathbf{v}, \mathbf{v}_j), \quad j = 1, \ldots, m \\ 0 : \text{else}, \end{cases} \tag{1}$$

where $d(\mathbf{v}, \mathbf{v}')$ is the distance in the visual descriptor space, usually the Euclidean distance $\|\mathbf{v} - \mathbf{v}'\|$. Summing the distributions of all features,

$$\mathbf{B}_I = \sum_{F \in \mathbf{F}_I} \mathbf{Q}_\mathbf{V}\mathbf{v}_F,$$

yields an affine-invariant representation of the image called a *bag of features*, which with proper normalization is a distribution of the image features over the visual vocabulary. Bags of features are often L_2-normalized and compared using the standard Euclidean distance or correlation, which allows efficient indexing and comparison using search trees or hash tables [11].

3 Spatially-Sensitive Image Descriptors

A major disadvantage of bags of features is the fact that they discard information about the spatial relations between features in an image. We are interested in *spatially-sensitive* bags of features that encode spatial information in an invariant manner. As already mentioned in the introduction, spatial information in the form of expressions is useful in disambiguating different uses of a word in text search. A 2D analogy of two text documents containing the same words up to some permutation is a scene depicting different arrangements or motion of the same objects: a change in the relative positions of the objects creates different spatial configuration of the corresponding features in the image. Yet, in images, the spatial relations can also change as a result of a difference in the view point (usually approximated by an affine transformation). If in the former case the difference in spatial relations is desired since it allows us to discriminate between different visual content, in the latter case, the difference is undesired since it would deem distinct a pair of visually similar images.

Visual expressions. A straightforward generalization of the notion of combinations of words and expressions to images can be obtained by considering *pairs* of features. For this purpose, we define a visual vocabulary on the space of pairs of visual descriptors as the product $\mathbb{V} \times \mathbf{V}$, and use the quantization operator $\mathbf{Q}_{\mathbf{V}}^2 = \mathbf{Q}_{\mathbf{V}} \times \mathbf{Q}_{\mathbf{V}}$ assigning to a pair of descriptors a distribution over $\mathbf{V} \times \mathbf{V}$. $(\mathbf{Q}_{\mathbf{V}}^2(\mathbf{v}, \mathbf{v}'))_{ij}$ can be interpreted as the joint probability of the pair $(\mathbf{v}, \mathbf{v}')$ being represented by the expression $(\mathbf{v}_i, \mathbf{v}_j)$.

Same way as expressions in text are pairs of adjacent words, visual expressions are pairs of spatially-close visual words. The notion of proximity can be expressed using the idea of canonical neighborhoods: fixing a disk M of radius $r > 0$ centered at the origin of the canonical system of coordinates, we define $N_F = \mathbf{A}_F M$ to be a *canonical neighborhood* of a feature F. Such a neighborhood is affine-covariant, i.e., $N_{\mathbf{T}F} = \mathbf{T}N_F$ for every affine transformation $\mathbf{T}$. The notion of a canonical neighborhood induces a division of pairs of features into near and far. We define a *bag of pairs of features* simply as the distribution of near pairs of features,

$$\mathbf{B}_I^2 = \sum_{F \in \mathbf{F}_I} \sum_{F' \in N_F} \mathbf{Q}_{\mathbf{V}}^2(\mathbf{v}_F, \mathbf{v}_{F'}).$$

Bags of pairs of features are affine-invariant by their construction, provided that the feature transform and the canonization are affine-covariant.

Spatial relations. Canonical neighborhoods express binary affine-invariant proximity between features, which is a simple form of spatial relations. A more general class of spatial relations can be obtained by considering the relation between the canonical transformations of pairs of features. Specifically, we consider the *canonical relation*

$$\mathbf{S}_{F,F'} = \mathbf{A}_{F'}^{-1} \mathbf{A}_F.$$

It is easy to show that $\mathbf{S}_{F,F'}$ is affine-invariant, i.e., $\mathbf{S}_{\mathbf{T}F,\mathbf{T}F'} = \mathbf{S}_{F,F'}$ for every affine transformation $\mathbf{T}$. This spatial relation can be thought of as the transformation from F' to F expressed in the canonical system of coordinates. It should not be confused with the transformation from the system of coordinates of F' to the system of coordinates of F, which is achieved by $\mathbf{A}_F \mathbf{A}_{F'}^{-1}$.

It is worthwhile noting that symmetric features result in ambiguous spatial relations. The problem can be resolved by projecting the relation onto the subgroup of the affine group modulo the ambiguity group. When the ambiguity group is finite (e.g. reflection), the spatial relation can be defined as a set [28].

Spatially-sensitive bags of features. Being an invariant quantity, the canonical spatial relation can be used to augment the information contained in visual descriptors in a bag of pairs of features. For that purpose, we construct a vocabulary of spatial relations, $\mathbf{S} = \{\mathbf{S}_1, \ldots, \mathbf{S}_n\}$. A quantization operator $\mathbf{Q}_{\mathbf{S}}$ associated with the spatial vocabulary can be constructed by plugging an appropriate metric into (1). The easiest way of defining a distance on the space of transformations is the Frobenius norm on transformations represented in homogeneous coordinates,

$$d^2(\mathbf{S},\mathbf{S}') = \|\mathbf{S}-\mathbf{S}'\|_{\mathrm{F}}^2 \;=\; \mathrm{tr}((\mathbf{S}-\mathbf{S}')^{\mathrm{T}}(\mathbf{S}-\mathbf{S}')),$$

which is equivalent to considering the 3×3 transformation matrices as vectors in $\mathbb{R}^9$ using the standard Euclidean distance. A somewhat better approach is to use the intrinsic (geodesic) distance on the Lie group of matrices,

$$d^2(\mathbf{S},\mathbf{S}') = \|\log(\mathbf{S}^{-1}\mathbf{S}')\|_{\mathrm{F}}^2,$$

where $\log \mathbf{X} = \sum_{i=0}^{\infty} \frac{(-1)^{i+1}}{i}(\mathbf{X}-\mathbf{I})^i$ is the matrix logarithm.

The disadvantage of the intrinsic distance is the non-linearity introduced by the logarithm. However, using the Baker-Campbell-Hausdorff exponential identity for non-commutative Lie groups yields the following first-order approximation,

$$\begin{aligned} d(\mathbf{S},\mathbf{S}') = \|\log(\mathbf{S}^{-1}\mathbf{S}')\|_{\mathrm{F}} \;&= \left\|\log\left(\exp(-\log\mathbf{S})\exp(\log\mathbf{S}')\right)\right\|_{\mathrm{F}} \\ &= \left\|\log\left(\exp(\log\mathbf{S}') - \exp(\log\mathbf{S}) + \mathcal{O}(\|\log\mathbf{S}'\log\mathbf{S}\|^2)\right)\right\|_{\mathrm{F}} \\ &\approx \|\log\mathbf{S}' - \log\mathbf{S}\|_{\mathrm{F}}. \end{aligned}$$

Practically, using this approximation, spatial relations can be thought of as nine-dimensional vector whose elements are the entries of the logarithm matrix $\log\mathbf{S}$, and the distance between them is the standard Euclidean distance on $\mathbb{R}^9$. A more general distance between spatial relations can be obtained by projecting $\mathbf{S}$ and $\mathbf{S}'$ onto subgroups of the affine group, measuring the distances between projections, and then combining them into a single distance.

Coupling the spatial vocabulary $\mathbf{S}$ with the visual vocabulary $\mathbf{V} \times \mathbf{V}$ of pairs of features, we define the *spatially-sensitive bag of features*

$$\mathbf{B}_I^3 = \sum_{F\in\mathbf{F}_I}\sum_{F'\in N_F} \mathbf{Q}_{\mathbf{V}}^2(\mathbf{v}_F,\mathbf{v}_{F'})\cdot\mathbf{Q}_{\mathbf{S}}(\mathbf{S}_{F,F'}),$$

which, with proper normalization, is a distribution over $\mathbf{V}\times\mathbf{V}\times\mathbf{S}$ that can be represented as a three-dimensional matrix of size $m \times m \times n$. Spatially-sensitive bags of features are again affine-invariant by construction.

While the clear advantage of spatially-sensitive bags of features is their higher discriminativity, the resulting representation size may be significantly higher. Additional potential drawback is that repeatability of pairs of features can be lower compared to single features. Due to the above considerations, the best application for the presented approach is a scenario in which the two images to be compared have a large overlap in the visual content. An example of such an application is image and video copy detection, in which one tries to recognize an image or video frame that has undergone some processing or tampering. Another example is video alignment, in which one tries to find a correspondence between two video sequences based on their visual content. Subsequent frames in video may differ as a result from motion, which result in different spatial configurations of the depicted object. Distinguishing between such frames using bags of features would be very challenging or even impossible (see e.g. Figure 2).

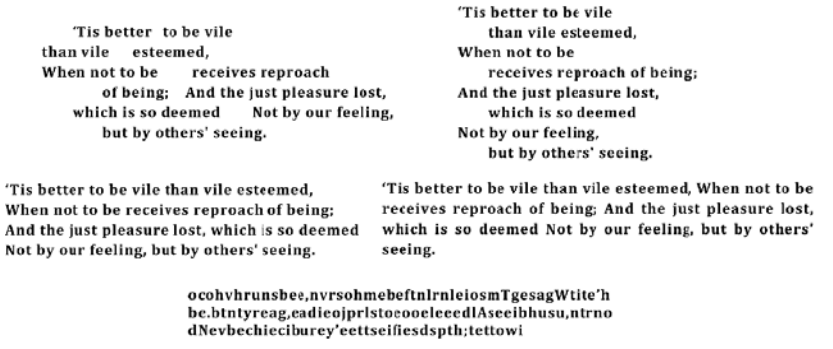


Fig. 1. Examples of five layouts of a Shakespearean sonnet from the *Text* dataset. The last layout is a random permutation of letters.

Fig. 2. Examples of three images from the same scene in the *Opera* dataset. Each scene contains visually similar objects appearing in different spatial configurations. Such images are almost indistinguishable by means of bags of features, yet, result in different spatially-sensitive descriptors.

4 Results

We assessed the proposed methods in three image retrieval experiment, using *Text*, *Opera*, and *Still life* datasets described in the following.[1] The first two experiments were with synthetic transformations, the third experiment was with real photographed data. The datasets were created to contain objects in different geometric configurations. In all the experiments, MSER was used as the feature detector, followed by the moment-based canonization. Feature descriptors were created by sampling the unit square in the canonical space on a 12×12 grid. Three methods were compared: simple bags of features (BoF), bags of pairs of features (P-BoF), and spatially-sensitive bags of features (SS-BoF). All bags of features were computed from the same sets of feature descriptors and canonical transformations using the same visual vocabularies.

Synthetic data. The first two experiments were performed on two datasets. The first was the *Text* dataset consisting of 29 distinct fragments from

[1] All the data and code for reproducing the experiments will be published online.

Shakespearian sonnets. Each fragment was rendered as a black-and-white image using the same font in several spatial layouts containing the same letters organized differently in space. One of such extreme layouts included a random permutation of the letters. This resulted in a total of 91 images, a few examples of which are depicted in Fig. 1. Black-and-white text images are an almost ideal setting for the MSER descriptor, which manifested nearly perfect affine-invariance. This allowed to study in an isolated manner the contribution of spatial relations to bag of feature discriminativity.

The *Opera* dataset was composed of 28 scenes from different opera recordings. From each scene, several frames were selected in such a way to include approximately the same objects in different spatial configurations, resulting in a total of 83 images (Fig. 2). The challenge of this data was to be able to distinguish between different spatial configurations of the objects. Such a problem arises, for example, in video alignment where subsequent frames are often very similar visually but have slightly different spatial layouts.

To each image in both data sets, 21 synthetic transformation were applied. The transformations were divided into five classes: in-plane rotation, mixed in-plane and out-of-plane rotation, uniform scaling, non-uniform scaling, and null (no transformation). Each transformation except the null appeared in three increasing strengths (marked $1-5$).

For the *Text* data, the vocabularies were trained on examples of other text, not used in the tests. Same visual vocabulary of size 128 were used in all the algorithms; spatial vocabulary of size 24 was used in SS-BoFs. For the *Opera* data, the vocabularies were trained on web images. Visual vocabulary was of size 128, and spatial vocabulary was of size 24. In all experiments, the size of the canonical neighborhood was set to $r = 15$.

We performed a leave-one-out retrieval experiment on both datasets. Euclidean distance between different image descriptors (BoF, P-BoF, and SS-BoF) was used to rank the results. Retrieval performance was evaluated on subsets of the distance matrix using precision/recall characteristic. *Precision at* k, $P(k)$, is defined as the percentage of relevant images in the first k top-ranked retrieved images. Relevant images were the same configuration of objects regardless of transformation. *Average precision* (AP) is defined as $mAP = \frac{1}{R}\sum_k P(k)\cdot rel(k)$, where $rel(k) \in \{0,1\}$ is the relevance of a given rank and R is the total number of relevant images. *Mean average precision* (mAP), the average of AP over all queries, was used as a single measure of retrieval performance. Ideal retrieval results in all first matches relevant (mAP=100%).

Tables 1 and 2 shows the retrieval performance using different image representations on *Text* and *Opera* datasets, respectively. The performance is broken down according to transformation classes and strengths. The use of spatially-sensitive bags of features increases the performance from 39.46% mAP to 92.4% (134% improvement) on the *Text* data and from 83.9% to 91.35% (8% improvement) on the *Opera* data.

Real data. In the third experiment, we used the *Still life* dataset containing 191 images of objects laid out in 9 different configurations (scenes) and captured

Table 1. Retrieval performance (mAP in %) of different methods on the *Text* dataset, broken down according to transformation classes and strengths (1–5)

		Strength				
Method	**Transformation**	**1**	**≤2**	**≤3**	**≤4**	**≤5**
BoF	*In-plane rotation*	41.57	35.33	31.98	30.86	30.39
	Mixed rotation	26.28	35.23	32.68	28.56	24.25
	Nonuniform scale	58.13	59.70	59.25	60.11	58.63
	Uniform scale	55.30	48.64	46.79	45.77	44.57
	All	45.32	44.73	42.68	41.33	39.46
P-BoF	*In-plane rotation*	60.51	49.36	43.45	40.90	39.94
	Mixed rotation	30.08	48.86	42.97	36.05	30.33
	Nonuniform scale	81.90	82.90	83.13	83.26	81.31
	Uniform scale	78.91	72.56	73.06	69.82	67.73
	All	62.85	63.42	60.65	57.51	54.83
SS-BoF	*In-plane rotation*	100.00	100.00	99.45	99.08	99.12
	Mixed rotation	97.99	98.99	98.14	85.96	70.48
	Nonuniform scale	100.00	100.00	100.00	100.00	100.00
	Uniform scale	100.00	100.00	100.00	100.00	100.00
	All	99.50	99.75	99.40	96.26	92.40

Table 2. Retrieval performance (mAP in %) of different methods on the *Opera* dataset, broken down according to transformation classes and strengths (1–5)

		Strength				
Method	**Transformation**	**1**	**≤2**	**≤3**	**≤4**	**≤5**
SS-BoF	*In-plane rotation*	92.95	88.36	84.62	80.63	77.59
	Mixed rotation	68.39	64.98	69.86	70.25	70.07
	Nonuniform scale	95.50	96.01	95.90	95.22	94.47
	Uniform scale	96.73	95.16	94.78	94.31	93.47
	All	88.39	86.13	86.29	85.10	83.90
SS-BoF	*In-plane rotation*	93.55	88.19	85.82	84.17	81.49
	Mixed rotation	75.42	72.84	75.47	75.21	74.75
	Nonuniform scale	95.31	96.01	95.53	95.13	94.52
	Uniform scale	96.18	94.76	93.86	93.62	93.09
	All	90.11	87.95	87.67	87.03	85.96
SS-BoF	*In-plane rotation*	95.11	92.73	91.27	89.91	88.52
	Mixed rotation	82.68	81.42	83.65	83.95	84.23
	Nonuniform scale	97.32	97.32	97.64	97.36	96.47
	Uniform scale	98.80	98.11	97.28	96.66	96.20
	All	93.48	92.40	92.46	91.97	91.35

from multiple views with very wide baseline by cameras with different focus and resolution (12–36 views for each scene). Some of the views differed dramatically, including occlusions, scene clutter, as shown in Figure 3. Moreover, most of the scenes included a sub-set of the same objects. The challenge in this experiment was to group images into scenes based on their visual similarity. Same

Fig. 3. Examples of three viewpoints (left, middle, right) and two configurations (first and second rows) of objects in the Still life dataset. Images in the same layout were taken by multiple cameras from different positions.

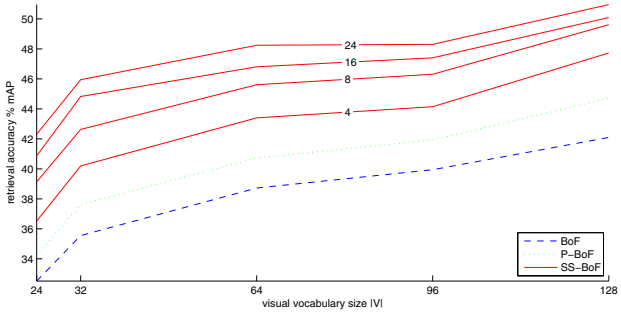


Fig. 4. Performance of different methods on the *Still life* dataset. Four red solid curves correspond to SS-BoF with spatial vocabulary of different size (displayed on the curve).

vocabularies as in the *Opera* test were used. We performed a leave-one-out retrieval experiment. Successful match was from the same object configuration (e.g., in Figure 3, when querying the top left image, correct matches are top middle and right, incorrect matches are all images in the second row).

Figure 4 shows the retrieval accuracy of different methods as a function of visual and spatial vocabulary size. BoF achieves the best retrieval performance (42.1% mAP) with a vocabulary of size 128. With the same vocabulary, P-BoF achieves 44.7% mAP. The best result for SS-BoF is 51.0% (21% improvement) when using a spatial vocabulary of size 24. Consistent and nearly constant improvement is exhibited for all the range of the tested visual vocabulary sizes.

We observe that while consistent improvement is achieved on all datasets, spatially-sensitive bags of features perform the best on the *Text* data. We attribute this in part to the relatively primitive feature canonization method used in our experiments, which was based only on the feature shape and not on the feature intensity content. This might introduce noise into the computed canonical transformation and therefore degrade the performance of canonical neighbors and spatial vocabulary. In future studies, we intend to use a SIFT-like feature canonization based on the dominant intensity direction, which is likely to improve the stability of the canonical transformations.

5 Conclusions and Future Directions

We presented a construction of a feature-based image representation that generalizes the bag of features approach by taking into consideration spatial relations between features. Central to our construction is a vocabulary of pairs of affine-invariant features coupled with a vocabulary of affine-invariant spatial relations. The presented approach is a meta-algorithm, since it augments the standard bag of features approach and is not limited to a specific choice of a feature transform. In future studies, we intend to test it on other descriptors such as SIFT, and extend the idea of spatial relations to epipolar relations between features in calibrated images. We also intend to extend the proposed approach to video, creating affine-invariant vocabularies for motion.

Experimental results show improved performance of image retrieval on synthetic and real data. We plan to evaluate our approach in a large-scale image retrieval experiment. Our approach is especially suitable for problems in which the compared images have large overlap in visual content, such as copy detection and video alignment, an application that will be studied in future works.

References

1. Lindeberg, T.: Feature detection with automatic scale selection. IJCV 30, 79–116 (1998)
2. Mikolajczyk, K., Schmid, C.: Indexing based on scale invariant interest points. In: Proc. ICCV., vol. 1, pp. 525–531 (2001)
3. Lowe, D.: Distinctive image features from scale-invariant keypoint. IJCV (2004)
4. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. IJCV 60, 63–86 (2004)
5. Tuytelaars, T., Van Gool, L.: Matching widely separated views based on affine invariant regions. IJCV 59, 61–85 (2004)
6. Kadir, T., Zisserman, A., Brady, M.: An affine invariant salient region detector. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 228–241. Springer, Heidelberg (2004)
7. 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)
8. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.: A comparison of affine region detectors. IJCV 65, 43–72 (2005)

9. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, p. 404. Springer, Heidelberg (2006)
10. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: Proc. CVPR (2003)
11. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: Proc. ICCV (2007)
12. Shechtman, E., Irani, M.: Matching local self-similarities across images and videos. In: Proc. CVPR, pp. 1–8 (2007)
13. Marszaek, M., Schmid, C.: Spatial weighting for bag-of-features. In: Proc. CVPR., vol. 2 (2006)
14. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: Workshop on Statistical Learning in Computer Vision, ECCV, pp. 17–32 (2004)
15. Grauman, K., Darrell, T.: Efficient image matching with distributions of local invariant features. In: Proc. CVPR., vol. 2 (2005)
16. Rubner, Y., Tomasi, C., Guibas, L.: The earth mover's distance as a metric for image retrieval. IJCV 40, 99–121 (2000)
17. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proc. CVPR (2006)
18. Wu, Z., Ke, Q., Isard, M., Sun, J.: Bundling features for large-scale partial-duplicate web image search. In: Proc. CVPR (2009)
19. Sivic, J., Zisserman, A.: Video data mining using configurations of viewpoint invariant regions. In: Proc. CVPR (2004)
20. Sivic, J., Russell, B., Efros, A., Zisserman, A., Freeman, W.: Discovering object categories in image collections. In: Proc. ICCV., vol. 2 (2005)
21. Chum, O., Matas, J.: Geometric hashing with local affine frames. In: Proc. CVPR (2006)
22. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proc. CVPR., vol. 2 (2006)
23. Amores, J., Sebe, N., Radeva, P.: Context-based object-class recognition and retrieval by generalized correlograms. IEEE Trans. PAMI 29, 1818–1833 (2007)
24. Ovsjanikov, M., Bronstein, A.M., Bronstein, M.M., Guibas, L.: Shape google: a computer vision approach to invariant shape retrieval. In: Proc. NORDIA (2009)
25. Behmo, R., Paragios, N., Prinet, V.: Graph commute times for image representation. In: Proc. CVPR (2008)
26. Forssén, P., Lowe, D.: Shape descriptors for maximally stable extremal regions. In: Proc. ICCV, pp. 59–73 (2007)
27. Muse, P., Sur, F., Cao, F., Lisani, J.L., Morel, J.M.: A theory of shape identification (2005)
28. Bronstein, A.M., Bronstein, M.M.: Affine-invariant spatial vocabularies. Technical Report Techn. Report CIS-2009-10, Dept. of Computer Science, Technion, Israel (2009)

Object Classification Using Heterogeneous Co-occurrence Features

Satoshi Ito and Susumu Kubota

Corporate Research & 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) [1], 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

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 209–222, 2010.

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

compute a gradient orientation image G from I
initialize co-occurrence histograms H with zeros
for $i = 1$ to m **do**
  for $j = 1$ to n **do**
    for all $(x, y) \in D_j$ **do**
      if $(x + p_i, y + q_i)$ is inside of the image **then**
        $g_1 \leftarrow G(x, y)$
        $g_2 \leftarrow G(x + p_i, y + q_i)$
        $H(g_1, g_2, j, i) \leftarrow H(g_1, g_2, j, i) + 1$
      end if
    end for
  end for
end for
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 θ and $\theta + 180$ degrees are equivalent and the orthogonal directions θ 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 θ 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 2×2 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

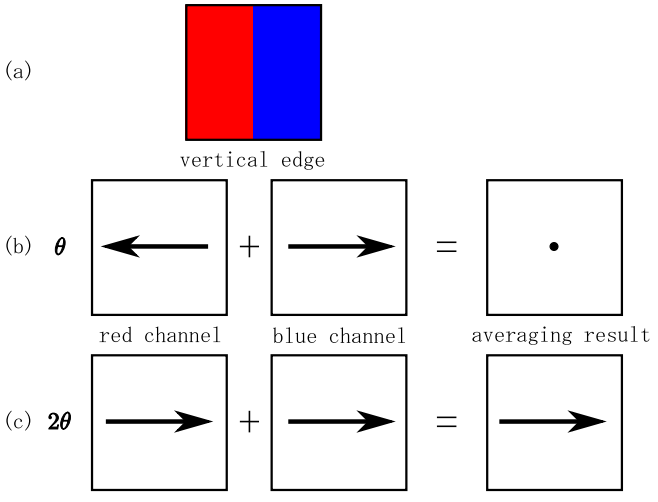


Fig. 1. (a) A vertical edge. (b) Averaging gradients, denoted by arrows, over red and blue channels in the single angle representation gives undesirable result. (c) Averaging gradients over red and blue channels in the double angle representation gives desirable result.

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 plane is as follows;

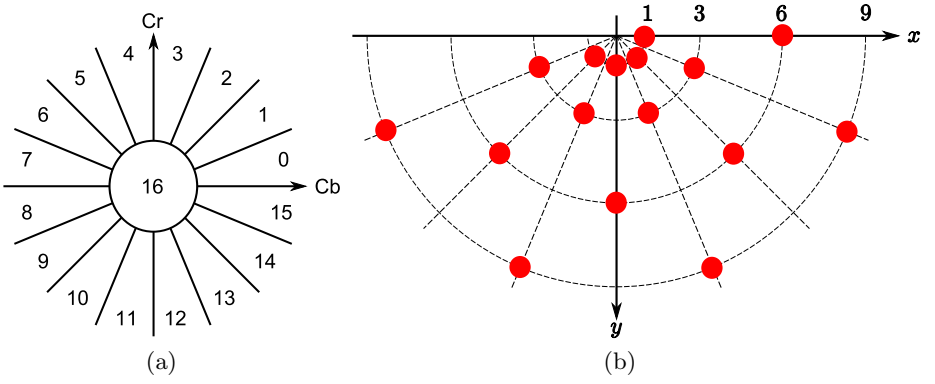


Fig. 2. (a) The figure shows color labels in Cb-Cr space. The label 16 corresponds to neutral gray. (b) The figure shows 16 offsets (drawn as *filled circles*) used for color-CoHOG calculation.

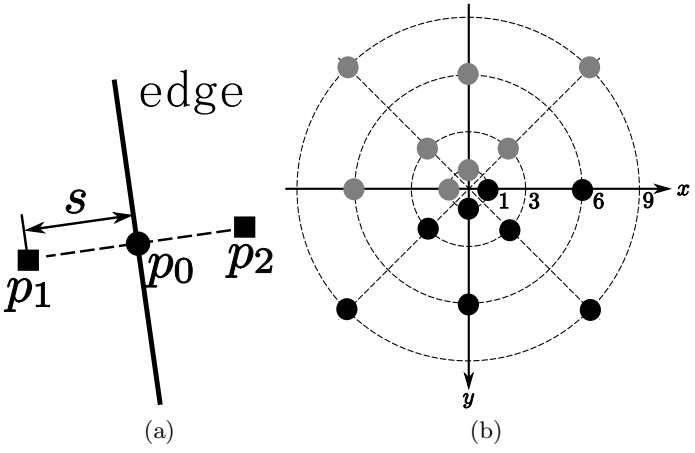
Algorithm 2. Color-CoHOG calculation

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

compute an edge direction image G from I using the double angle representation
compute color labels C of pixels in the image
initialize co-occurrence histograms H with zeros
for $i = 1$ to m **do**
  for $j = 1$ to n **do**
    for all $(x, y) \in D_j$ **do**
      if $(x + p_i, y + q_i)$ is inside of the image **then**
        $g_1 \leftarrow G(x, y)$
        $g_2 \leftarrow G(x + p_i, y + q_i)$
        if $C(x, y)$ is equal to $C(x + p_i, y + q_i)$ **then**
          $c \leftarrow 1$
        else
          $c \leftarrow 0$
        end if
        $H(g_1, g_2, c, j, i) \leftarrow H(g_1, g_2, c, j, i) + 1$
      end if
    end for
  end for
end for
return H

$$H_{\mathrm{Y-Cb}}(g, c) \leftarrow H_{\mathrm{Y-Cb}}(g, c) + |dy| + |du| \tag{1}$$

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



(a) (b)

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.

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) $= 128$.

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

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

initialize co-occurrence histograms H with zeros
for $i = 1$ to m **do**
  for $j = 1$ to n **do**
    for all $(x, y) \in D_j$ **do**
      if $(x + p_i, y + q_i)$ and $(x - p_i, y - q_i)$ are inside of the image **then**
        $u_1 \leftarrow U(x + p_i, y + q_i) - U(x, y)$
        $v_1 \leftarrow V(x + p_i, y + q_i) - V(x, y)$
        $u_2 \leftarrow U(x - p_i, y - q_i) - U(x, y)$
        $v_2 \leftarrow V(x - p_i, y - q_i) - V(x, y)$
        $c_1 \leftarrow (u_1 > 0) + 2 \times (v_1 > 0)$ // quantization into 4 directions
        $c_2 \leftarrow (u_2 > 0) + 2 \times (v_2 > 0)$ // quantization into 4 directions
        $H(c_1, c_2, j, i) \leftarrow H(c_1, c_2, j, i) + 1$
      end if
    end for
  end for
end for
return H

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 64×128 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 4×8 non-overlapped small regions that are 16×16 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.

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

Fig. 4. Examples in the INRIA person dataset

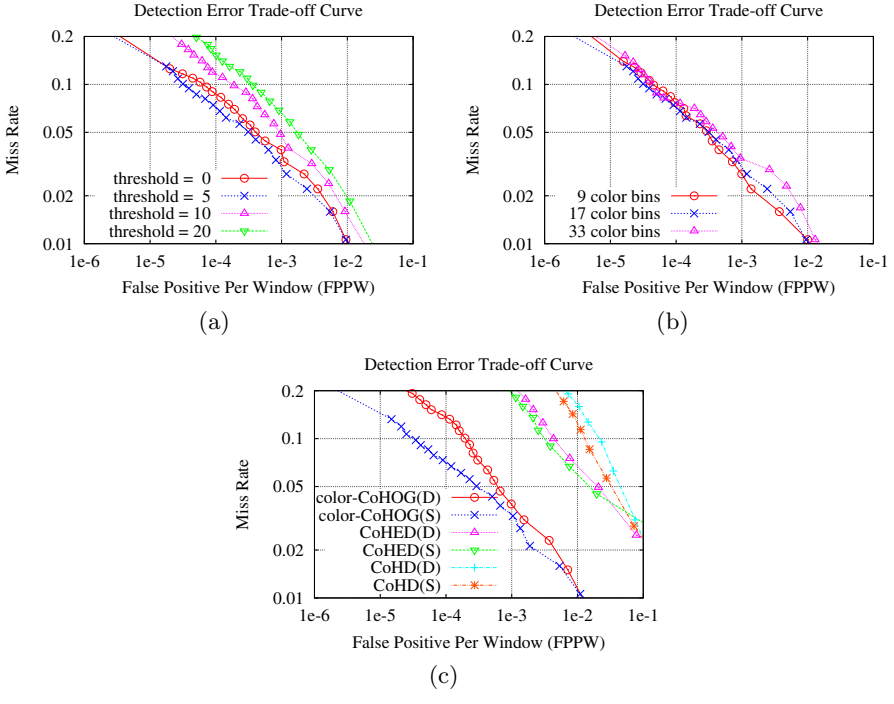


Fig. 5. Feature evaluation on the INRIA person dataset. (a) DET curves of various thresholds for neutral gray. (b) DET curves obtained by changing the number of color bins. (c) DET curves of the sparse setting (denoted by 'S') and the dense setting (denoted by 'D'), respectively.

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 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

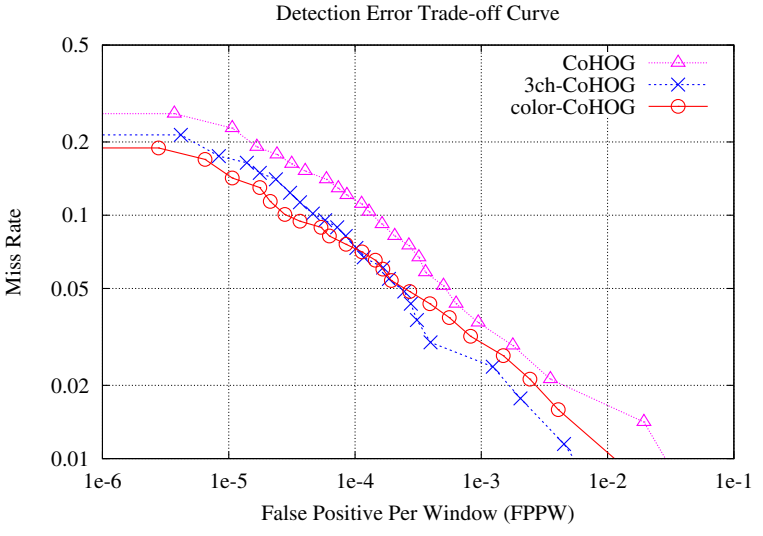


Fig. 6. DET curves of several CoHOGs on the INRIA person dataset. Color-CoHOG (*circle*) is superior to CoHOG (*triangle*) and comparable to 3ch-CoHOG (*cross*) while the dimensionality of color-CoHOG is half as that of CoHOG and only one-sixth of that of 3ch-CoHOG.

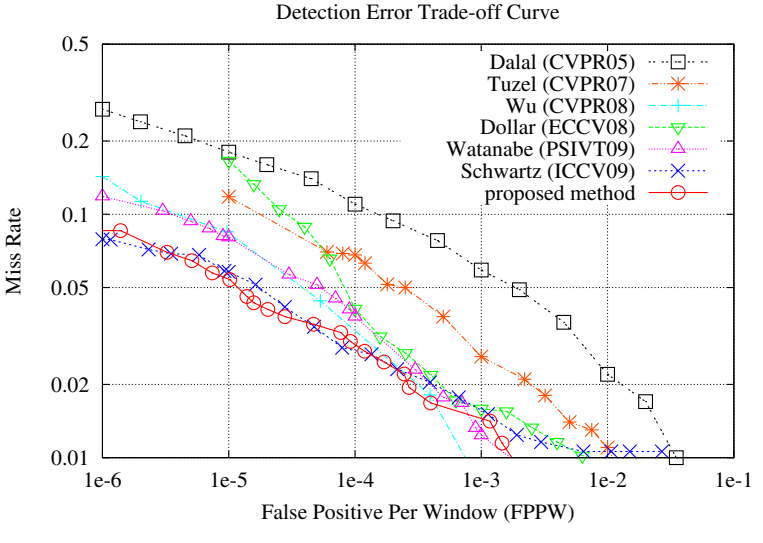


Fig. 7. DET curves of the proposed method and several previous methods on the INRIA person dataset. This figure shows that the proposed method (*circle*) is comparable to the state-of-the-art method (*cross*).

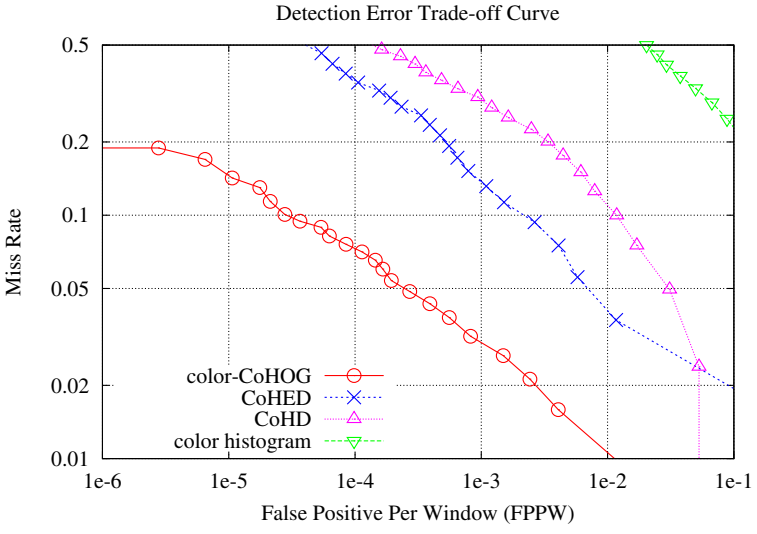


Fig. 8. DET curves of single features on the INRIA person dataset

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 64×64 pixel size. We extract color-CoHOG, CoHED, CoHD, and a color histogram from the whole region of the resized image and concatenate 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

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

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, Los Alamitos (1997)
7. Koschan, A.: A comparative study on color edge detection. In: Li, S., Teoh, E.-K., Mital, D., Wang, H. (eds.) ACCV 1995. LNCS, vol. 1035, pp. 574–578. Springer, Heidelberg (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: Proc. of Int. Conf. on Image Analysis and Recognition, Part I, pp. 753–760 (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)

Maximum Margin Distance Learning for Dynamic Texture Recognition

Bernard Ghanem and Narendra Ahuja

Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
{bghanem2,ahuja}@vision.ai.uiuc.edu

Abstract. The range space of dynamic textures spans spatiotemporal phenomena that vary along three fundamental dimensions: spatial texture, spatial texture layout, and dynamics. By describing each dimension with appropriate spatial or temporal features and by equipping it with a suitable distance measure, elementary distances (one for each dimension) between dynamic texture sequences can be computed. In this paper, we address the problem of dynamic texture (DT) recognition by learning linear combinations of these elementary distances. By learning weights to these distances, we shed light on how "salient" (in a discriminative manner) each DT dimension is in representing classes of dynamic textures. To do this, we propose an efficient maximum margin distance learning (MMDL) method based on the Pegasos algorithm [1], for both class-independent and class-dependent weight learning. In contrast to popular MMDL methods, which enforce restrictive distance constraints and have a computational complexity that is cubic in the number of training samples, we show that our method, called *DL-PEGASOS*, can handle more general distance constraints with a computational complexity that can be made linear. When class dependent weights are learned, we show that, for certain classes of DTs , spatial texture features are dominantly "salient", while for other classes, this "saliency" lies in their temporal features. Furthermore, *DL-PEGASOS* outperforms state-of-the-art recognition methods on the UCLA benchmark DT dataset. By learning class independent weights, we show that this benchmark does not offer much variety along the three DT dimensions, thus, motivating the proposal of a new DT dataset, called DynTex++.

1 Introduction

A dynamic texture (DT) sequence captures a stochastic spatiotemporal phenomenon. The randomness reflects in the spatial and temporal changes in the image signal. This may be caused by a variety of physical processes, e.g., involving objects that are small (smoke particles) or large (snowflakes), or rigid (grass, flag) or nonrigid (cloud, fire), moving in 2D or 3D, etc. Even though the overall global motion of a DT may be perceived by humans as being simple and coherent, the underlying local motion is governed by a complex stochastic model. For

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 223–236, 2010.

example, a scene of "translating" clouds conveys visually identifiable global dynamics; however, the implosion and explosion of the cloud segments during the motion result in very complicated local dynamics. Irrespective of the nature of the physical phenomena, the usual objective of DT modeling in computer vision and graphics is to capture the nondeterministic, spatial and temporal variation in images. The study of DTs poses numerous challenges, especially for traditional motion models that fail to capture their stochastic nature. These challenges arise from the need to capture the large number of objects involved, their complex motions, and their intricate interactions. A good model must accurately and efficiently capture both the appearance and global dynamics of a DT. Despite the diverse types of DTs in nature, we see that they belong to a three dimensional DT space. In this space, each dimension isolates a single aspect that describes the variation of an individual DT. These dimensions are, therefore, broad categories of variation for DTs, in general. However, they are not generally independent, since for some cases of DT, it is not possible to fix two dimensions and vary the third independently. This interdependence is attributed to the physical nature of the phenomena being imaged. In what follows, we will describe each of these dimensions and give their respective ranges. Then, we will designate the portion of the DT space, where this paper operates. Note that the first two dimensions describe the spatial variation and the spatial organization of a DT, while the third describes its temporal variations.

1. **Spatial Texture Element:** This dimension describes the spatial variation of a DT as observed from each frame independently. Texture elements (usually denoted as *texels*) are the spatially repetitive groups of pixels that share statistically similar appearance and structural properties. The spectrum of texture elements varies from the simplest form at the microscopic level (i.e. particles) to the most complex at the macroscopic level (i.e. whole objects). At one extreme, this spectrum has DTs that show clouds, smoke, or water in motion, while at the other, there are DTs of birds, animals, or humans moving. The majority of DT work has focused on pixel or subpixel objects (i.e. microscopic), whereby the pixel is assumed to be the texture element whose motion is to be modeled.
2. **Spatial Texture Layout:** This dimension describes the spatial layout of the texture elements in a DT, as well as, their spatial layering. A DT's spatial layout determines how its texture elements are organized within each frame, especially in terms of their spatial placement. In this sense, there are DTs with homogenously placed/spaced texture elements, as well as, DTs where the placement distribution is non-uniform. Moreover, the spatial layering of a DT refers to the "density" (or translucency) of a DT. For simplicity, spatial layering of a DT can be viewed as the alpha matte of the texture elements, in each frame, when visualized infront of a background layer. The values of this alpha matte take values in $[0, 1]$. For opaque DTs, spatial layering is not an issue, since the background does not appear at all (i.e. the alpha matte is either 0 or 1). For translucent DTs (e.g. clouds and smoke), this layering is essential. The majority of DT work has focused on DTs with opaque texture elements that cover the whole spatial extent of the video.

3. **Dynamics:** This dimension describes the temporal variation of a DT as observed by the frame-to-frame variation in its texture elements and their layering/layout. DT dynamics represent temporal changes in features (e.g. intensity values and linear transformations of these values) describing the texture elements and their layout. Note that the dynamics of a DT is a global motion representation that incorporates the dynamics of individual texture elements and their spatiotemporal interactions. Being a DT means that the dynamics of texture elements are statistically similar and temporally stationary. In other words, texture elements in the same DT all "move" in a similar fashion and their "motions" are not time dependent (i.e. statistically stationary). As such, the models of DT dynamics either make use of physical models (e.g. Navies-Stokes equations [2]) or assume a general parametric model whose parameters are learned by fitting the model to the observed DT frames (e.g. a linear dynamical system [3]). The majority of DT work has concentrated on the latter form of models, where linear/nonlinear models have been proposed to model variations in the intensity values of DTs.

In this paper, we cater to opaque DTs consisting of pixel-based texture elements, whose dynamics can be represented by a linear parametric model [3]. We address the problem of DT recognition, which is motivated by critical real-life applications, especially the detection of the onset of emergencies (e.g. fire). Recognition is done by learning linear combinations of distances between DT sequences, so that classes of DTs are maximally separated. These distances quantify how different two DT sequences are with respect to the three dimensions mentioned above. By learning weights to these distances, we shed light on how "salient" (in a discriminative fashion) each dimension (i.e. spatial and/or temporal) is in representing a single DT class or a whole DT database.

2 Related Work

DT recognition involves the analysis of both image appearance and temporal changes in appearance. For an overview of recent techniques developed for DT recognition, we refer the reader to [4]. Numerous DT recognition methods have stemmed from representing the global spatiotemporal variations of a DT as a linear dynamical system (LDS) [3]. In [5], Doretto et al. use the LDS model parameters and the Martin distance measure [6] to perform nearest neighbor recognition. In [7], a kernel function between two LDS models was proposed and used in a support vector machine (SVM) framework to perform DT recognition. More recent work has addressed shift and view invariant DT recognition [8,9]. The latter work extends the use of the popular bag-of-features model to the non-Euclidean space of LDS models.

Other recognition methods have used a multiplicity of spatiotemporal descriptors to represent a DT sequence. In [10], Peteri et al. propose a DT recognition algorithm based on six translation invariant features. Recent work by Zhao et al. proposed using local binary patterns (LBP) [11] and volume local binary patterns (VLBP) to recognize DT sequences [12,13]. The latter two methods are

based on local descriptors, which do not incorporate the global dynamics that characterize a DT.

Despite the merits of these methods, they all either focus on one dimension of the DT space defined before or assume that these dimensions contribute equally and in the same manner for all DT classes. These assumptions are quite restrictive and fail to characterize the discriminative properties of many DTs. To the best of our knowledge, this paper is the first to address the problem of combining the discriminative properties of the three DT dimensions. Here, we provide an intuitive example that motivates why this is important in DT recognition. On one hand, the fire DT class is easily distinguished from other DT classes, primarily due to its highly discriminative dynamics, as compared to its spatial texture appearance. On the other hand, DTs such as moving leaves and grass have a more "salient" spatial texture element.

We infer the contributions of the DT dimensions by using a multiplicity of DT descriptors, each of which operates in a given dimension. We elaborate on these descriptors and motivate their selection later. Since these descriptors are of different dimensions and belong to different spaces, we model the distance between two DT sequences as a weighted sum of the elementary distances between their respective descriptors. Learning these weights in a maximum margin setting will determine the contributions of the DT dimensions, in such a way that maximizes DT class discrimination. Learning weighted distance functions in a maximum margin framework is not new, as it has been successfully applied to image classification and retrieval [14,15] and more recently to region-based object recognition [16]. These approaches impose the following distance constraint: an image is closer to all other images in its class than to images of all other classes. In feature space, this forces classes to be significantly compact, which tends not to be the case for most real data. This "compactness" assumption is quite restrictive and does not generalize well to object classes that share properties (e.g. cow vs. horse). Furthermore, this assumption produces a number of distance constraints/variables that is cubic in the number of training images, since all relevant distance triplets are used. Our method generalizes this "compactness" assumption whereby each DT sequence is only closer to a *representative* set of DTs within its class than to a *comparative* set of DTs outside this class. By taking the *representative* set of a DT to include its k nearest neighbors within its class and its *comparative* set to include all other DTs outside its class, we allow for less compact DT classes and much fewer distance constraints. To reduce computational complexity, we solve the primal version of the maximum margin problem in a way similar to the Pegasos algorithm [1].

Here, we note that distance weight learning finds some similarities with multiple kernel learning (MKL), which has been recently applied to object detection [17,18]. In MKL, the kernels define similarities between elements and are, by definition, symmetric and positive definite kernels. Although similarities can be formed from certain distances (e.g. by parametric negative exponentiation), these distances need not be symmetric and the parameters used to form the similarities need to be set wisely. This method also suffers from a computational drawback, since it requires expensive optimization techniques to learn the kernel mixing

coefficients. Moreover, the MKL framework does not readily accommodate the distance constraints required in maximum margin distance learning (MMDL).

Contributions: The contributions of this work are three fold. **(1)** We propose to learn the individual contributions/weights of all three DT dimensions, in regards to DT class discrimination. **(2)** To learn these weights, we propose an efficient MMDL method based on the Pegasos algorithm, whose complexity can be made linear in the number of training samples. **(3)** A new DT dataset, called DynTex++, is compiled to replace the current UCLA benchmark dataset.

This paper is organized as follows. In Section 3, we give an overview of the DT recognition problem, in an MMDL framework. Section 4 provides a detailed description of our proposed solution and algorithm, while Section 5 shows experimental validation of this algorithm, when applied to the UCLA and DynTex++ datasets.

3 Problem Overview

In this paper, we seek to learn how the different dimensions of the DT space can be linearly combined to best discriminate between DT classes. Learning these linear combinations for a given DT class or a group of DT classes sheds light on the relative importance of each DT dimension. We choose a suitable descriptor to represent each dimension, which is characterized by a corresponding elementary distance. Since these descriptors need not belong to vector spaces, the elementary distances are can be of different forms. In this framework, the distance between two DT sequences is modeled as a positively weighted sum of their elementary distances. These weights are learned in a maximum margin fashion, so that DT classes are maximally separated. We consider the case of class independent and class dependent weights.

We assume a set of M training DT sequences (from N classes) is given with corresponding labels in $\{1, \ldots, N\}$. Let $\ell(.)$ denote the labeling function, whereby $\ell(v_i)$ is the label of the DT sequence v_i. The DT sequence v_i has F different DT descriptors[1], which characterize the three different DT dimensions. We define the f^{th} elementary distance from v_j to v_i as $d_f(v_i \rightarrow v_j)$. Here, we note that these elementary distances need not be symmetric. As such, the combined distance from v_j to v_i is defined as $D_{\boldsymbol{w}_{\ell(v_i)}}(v_i \rightarrow v_j) = \sum_{f=1}^{F} w^f_{\ell(v_i)} d_f(v_i \rightarrow v_j)$. More compactly, we can combine the elementary distances in vector format to obtain $D_{\boldsymbol{w}_{\ell(v_i)}}(v_i \rightarrow v_j) = \boldsymbol{w}^T_{\ell(v_i)} \boldsymbol{d}(v_i \rightarrow v_j)$. Here, $w^f_{\ell(v_i)}$ is the weight that characterizes the f^{th} elementary distance for class $\ell(v_i)$. Here, we are considering class dependent weights; however, class independent weights are similarly incorporated by dropping the class label from $\boldsymbol{w}_{\ell(v_i)}$.

In order to best separate the DT classes, we assume that each DT of a given class is closer to a *representative* set of DTs within this class than a *comparative* set of DTs outside this class. Let $\mathcal{R}(v_i)$ define the *representative* set corresponding to DT v_i and $\mathcal{C}(v_i)$ define its *comparative* set. Under this assumption,

[1] In this paper, $F = 3$, but the method generalizes to any number of descriptors.

a set of distance constraints arises for each DT v_i, defined as follows. For all $i \neq j, \ell(v_i) = \ell(v_j) \neq \ell(v_k)$, $v_j \in \mathcal{R}(v_i)$, and $v_k \in \mathcal{C}(v_i)$ we have:

$$D_{\boldsymbol{w}_{\ell(v_i)}}(v_i \rightarrow v_j) \leq D_{\boldsymbol{w}_{\ell(v_i)}}(v_i \rightarrow v_k) \Leftrightarrow \boldsymbol{w}_{\ell(v_i)}^T \triangle \boldsymbol{d}(v_i, v_j, v_k) \geq 0 \qquad (1)$$

where $\boldsymbol{d}(v_i, v_j, v_k) = \boldsymbol{d}(v_i \rightarrow v_k) - \boldsymbol{d}(v_i \rightarrow v_j)$ is the distance difference corresponding to the DT triplet v_i, v_j, and v_k. The total number of these constraints is $\sum_i^M |\mathcal{R}(v_i)||\mathcal{C}(v_i)|$. Clearly, this number and thus the scale of the optimization needed to learn $\boldsymbol{w}_{\ell(v_i)}$ depends on the nature of $\mathcal{R}(.)$ and $\mathcal{C}(.)$. In fact, it is bounded by $\Theta(M^3)$ from above and $\Theta(M)$ from below.

Let $\mathbf{A}_c \in \mathbb{R}^{L \times F}$ denote the matrix whose rows are composed of all the distance difference vectors $\triangle \boldsymbol{d}(v_i, v_j, v_k)$ for all DTs v_i where $\ell(v_i) = c$. The distance constraints in Eq. (1) can be formalized as $\mathbf{A}_c \boldsymbol{w}_c \succeq \mathbf{0}$. We embed these constraints in a maximum margin framework, as shown in Eq. (2). In this framework, the cost function includes two terms that work towards minimizing the classification bias and variance. The second term is the average hinge loss cost of the L distance constraints. This cost uses a margin of 1 instead of 0. Although using L_1 regularization is known to lead to sparser solutions, we choose an L_2 regularization term on $\boldsymbol{w}_c$ instead, as it is more robust to noise and outliers and the number of feature descriptors F is relatively too small to benefit from a sparse solution.

$$\min_{\boldsymbol{w}_c \succeq \mathbf{0}} \frac{\lambda}{2} \|\boldsymbol{w}_c\|_2^2 + \frac{1}{L} \sum_{i=1}^{L} \max\left(0, 1 - \boldsymbol{w}_c^T \boldsymbol{a}_c(i)\right) \qquad (2)$$

where $\boldsymbol{a}_c(i)$ is the i^{th} row in $\mathbf{A}_c$. It is important to point out that when solving for class independent weights the matrix of distance constraints becomes a concatenation of all $\mathbf{A}_c$ matrices with $c \in \{1, \ldots, N\}$. Furthermore, note that class information need not be provided so long as relative dissimilarities/rankings are. In other words, even when class labels are not given, our method can still be applied, if pairwise distance inequalities are known. So, a statement like "dynamic texture A looks more similar to dynamic texture B than C" can be directly translated to a distance constraint.

The formulation in Eq. (2) is the same one used in the Pegasos algorithm [1], except for the non-negativity constraint on $\boldsymbol{w}_c$. In the next section, we will show how the original Pegasos method can be modified to efficiently solve for $\boldsymbol{w}_c$, to incorporate different forms of $\mathcal{R}(.)$ and $\mathcal{C}(.)$, and to reduce the number of distance constraints used in each Pegasos iteration. In fact, we choose to use this formulation/method instead of the one used in [14,15,16], since the latter does not lend itself suitable for variations in the *representative* and *comparative* sets and it requires a custom solver to handle a large number of distance constraints.

After solving for $\boldsymbol{w}_c$ of each class, a test DT sequence is classified as the class, which satisfies the most (or violates the least) number of distance constraints generated by the test DT. More specifically, for each class in the training set, a logistic regression classifier[2] is learned based on the combined distances of

[2] For a test sequence v_p, $f(v_p|c) = \left(1 + \exp\left(\alpha_0 + \sum_{v_i:\ell(v_i)=c} \alpha_i D_{\boldsymbol{w}_c}(v_i \rightarrow v_p)\right)\right)^{-1}$ defines the logistic regression classifier of class c.

training samples to samples within this class, as done in [16]. The test DT is assigned to the class, whose regression classifier evaluates to the maximum value among all classes. In the case of class independent weight learning, a simple k-nearest neighbor (kNN) classifier can be employed to classify the test DT.

Elementary Distances

In what follows, we present and justify the set of feature descriptors ($F = 3$) that we choose to represent the three DT dimensions of a DT sequence.

1. **Spatial Texture Element:** This DT dimension is described by a histogram of Local Binary Patterns (LBP), which provides a simple yet powerful local depiction of intensity variation. Each frame in a DT is described by an LBP histogram. As such, the elementary distance between two DTs along this dimension is the minimum distance between LBP histograms from these two DTs. To compare histograms, we use the Earth Mover's Distance (EMD) [19], which though more computationally expensive than other distances (e.g. ℓ_2 norm or χ^2), it provides a more accurate histogram distance. This spatial texture descriptor has been successfully utilized in DT recognition [12] and extended to video sequences in [20]. Recently, it has also proven to be useful in improving human detection performance [21].
2. **Spatial Texture Layout:** This DT dimension is described by a Pyramid of Histograms of Oriented Gradients (PHOG), which provide a powerful depiction of local spatial layout. In building the PHOG of a DT frame, we assume uniform weighting for each histogram at a given pyramid level and we normalize with respect to the number of histograms at each pyramid level. We only use two levels in the pyramid. Similar to the LBP descriptor, we use EMD to compute distances between histograms. Prior work has used this descriptor extensively in detecting objects, especially human detection [22], as well as, image retrieval [23].
3. **Dynamics:** To describe the global temporal variations of a DT sequence, we model it as a Linear Dynamical System (LDS) [3]. An LDS model is parameterized by the matrix pair $(\mathbf{A}, \mathbf{C})$, which govern feature generation and state transition. We assume a model size of 25, in our experiments. The LDS model and its variants have been extensively applied to DT recognition, most recently in [8,9]. The elementary distance between two LDS models is the Martin distance between ARMA processes [6].

Since each elementary distance above spans a different range of values, proper normalization is called for. After computing the elementary distances between DT sequences in the training set, we normalize each distance type by its mean (μ) offset by a multiple of its standard deviation (σ). In our experiments, we normalize each elementary distance by its corresponding ($\mu + 3\sigma$).

4 Learning Maximum Margin Weights

In this section, we give a detailed description of the learning algorithm used to compute $\boldsymbol{w}_c$ in Eq. (2). **Algorithm 1** summarizes the learning process, which

is a modified version of the original Pegasos algorithm [1]. *DL-PEGASOS* can handle general definitions for $\mathcal{R}(.)$ and $\mathcal{C}(.)$, since they can be data-driven and/or application specific. Furthermore, these definitions can even be dependent on $\boldsymbol{w}_c$, which explains why $\mathcal{R}(v_i)$ and $\mathcal{C}(v_i)$ must be updated at each iteration of this algorithm (refer to STEPS 2-3). In this case, Eq. (2) is no longer convex and so *DL-PEGASOS* becomes a stochastic, projected[3] subgradient descent method that alternates between **(i)** performing a Pegasos iteration when the sets $\mathcal{R}(v_i)$ and $\mathcal{C}(v_i)$ are fixed for all DT sequences v_i and **(ii)** updating these sets for a fixed Pegasos solution of $\boldsymbol{w}_c$. A study of convergence for *DL-PEGASOS* is kept for future work; however, empirical analysis is very promising.

Algorithm 1. *Distance Learning PEGASOS (DL-PEGASOS)*

Input : $\mathcal{R}(.)$, $\mathcal{C}(.)$, $\{\boldsymbol{d}(v_i \rightarrow v_j) : \ell(v_i) = c\}$, λ, T, m

Initialization: $\boldsymbol{w}_c^{(0)} \in B_\lambda^+ = \{\boldsymbol{x} : \|\boldsymbol{x}\|_2 \leq \frac{1}{\sqrt{\lambda}}, \boldsymbol{x} \succeq \boldsymbol{0}\}$

for $t = 0, \ldots, T$ **do**

- determine $\mathcal{R}(v_i)$ and $\mathcal{C}(v_i)$ $\forall v_i$ such that $\ell(v_i) = c$ (use $\boldsymbol{w}_c^{(t)}$ if needed)
- determine $\mathbf{A}_c \in \mathbb{R}^{L \times F}$

// original PEGASOS iteration

- Randomly choose $C_t \subseteq \{1, \ldots, L\}$, where $|C_t| = m$
- Set $C_t^+ = \{i \in C_t : \boldsymbol{a}_c^T(i)\boldsymbol{w}_c^{(t)} < 1\}$ and $\eta_t = \frac{1}{\lambda t}$
- Compute subgradient: $\boldsymbol{\nabla}_t = \lambda \boldsymbol{w}_c^{(t)} - \frac{1}{|C_t^+|} \sum_{i \in C_t^+} \boldsymbol{a}_c(i)$
- Do subgradient descent step: $\boldsymbol{w}_c^{(t+\frac{1}{2})} = \boldsymbol{w}_c^{(t)} - \eta_t \boldsymbol{\nabla}_t$
- Project onto B_λ^+: $\boldsymbol{w}_c^{(t+1)} = \min\left\{1, \frac{1/\sqrt{\lambda}}{\left\|\left[\boldsymbol{w}_c^{(t+\frac{1}{2})}\right]_+\right\|_2}\right\}\left[\boldsymbol{w}_c^{(t+\frac{1}{2})}\right]_+$

end

return $\boldsymbol{w}_c^{(T)}$

In our MMDL formulation, the distance constraint matrix $\mathbf{A}_c$ is directly dependent on the definition of $\mathcal{R}(.)$ and $\mathcal{C}(.)$. One popular definition is to equate $\mathcal{R}(v_i)$ to the set of all DTs within class c and $\mathcal{C}(v_i)$ to the set of all DTs outside class c (refer to Fig. 1(a)). This definition was used in [14,15,16]. This is quite restrictive, since it assumes that classes in feature space must be significantly compact (i.e. the minimum distance between any sample in class B to class A is at least the maximum distance between any two samples in class A). This is usually not the case for most real data. Based on this definition, the total number of distance constraints is $L = \Theta(M^3)$, which quickly becomes intractable for reasonably sized datasets. As a result, heuristic pruning measures were taken to

[3] The projection onto B_λ^+ is necessary due to the non-negativity constraint on $\boldsymbol{w}_c$. The $[.]_+$ operator returns a vector whose negative coordinates are truncated to zero.

reduce this number [15,14]; however, it remains $\Theta(M^3)$. A major problem with these measures is their immutability, since relevant constraints that are pruned at the beginning can never be added back to the learning process. Therefore, a need arises for another definition of $\mathcal{R}(.)$ and $\mathcal{C}(.)$ that is less restrictive (i.e. a more general representation of real data) and less computationally demanding.

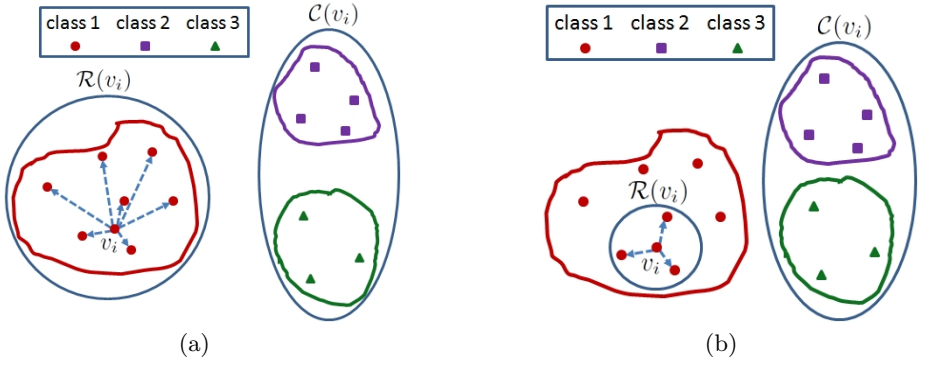


Fig. 1. Shows examples of two definitions for $\mathcal{R}(v_i)$ and their impact on the relative positioning of classes in feature space. For illustration purposes, we assume an L_2 distance is used between features. 1(a) is an example of the definition used in [14,15,16]. 1(b) is an example of the definition used here. Note how the classes need to be more separated (or equivalently more compact) in 1(a) than 1(b).

Although our MMDL method can handle a general structure for $\mathcal{R}(.)$ and $\mathcal{C}(.)$, in this paper, we set $\mathcal{R}(v_i)$ to the k nearest neighbors of v_i within its class. This is based on the intuition that a simple kNN classifier can be easily employed to classify v_i. In this case, STEPS 2-3 in **Algorithm 1** are equivalent to finding v_i's nearest neighbors according to $\boldsymbol{w}_c^{(t)}$. Note that the value of k need not be the same for every class c. A similar scheme can be applied to set $\mathcal{C}(v_i)$; however, since $M \gg N$ and to avoid overhead computation, we do not compute the nearest neighbors of v_i outside class c. Instead, we simply let $\mathcal{C}(v_i)$ be the set of all DTs outside class c (refer to Fig. 1(b)). Since $k \ll M$, the total number of distance constraints now is $L = \Theta(M^2)$. However, only m out of L constraints are actually used in a single iteration and m is usually much smaller than L. In fact, we show empirical results where the total number of constraints per *DL-PEGASOS* iteration can be reduced to $m = \Theta(M)$, without loss in recognition performance. Since a random set of these relevant constraints is chosen every iteration, the immutability problem facing previous methods is also alleviated. Moreover, the computational complexity of *DL-PEGASOS*, with $\mathcal{R}(.)$ and $\mathcal{C}(.)$ defined as above, is $\Theta\left(T(\frac{2F+k}{N}M + Fm)\right)$, which includes computing and sorting the combined distances $D_{\boldsymbol{w}_c}$. While previous MMDL methods suffer from $\Theta(M^3)$ complexity, our method is at worst $\Theta(M^2)$ and on average $\Theta(M)$.

5 Experimental Results

In this section, we present experimental results that validate the *DL-PEGASOS* algorithm[4] in terms of DT recognition. We first learn class-independent and class-dependent weights for the UCLA benchmark dataset [5]. Realizing that recognition performance on this dataset has saturated and that it lacks DT diversity, a new, easily accessible benchmark is essential. We organize the DynTex++ dataset to be this next benchmark and evaluate our algorithm on it.

5.1 UCLA Dataset

The UCLA dynamic texture dataset contains 50 classes of gray-scale dynamic texture, each of which is comprised of 4 DT sequences. Since these 50 classes contain the same DTs at different viewpoints, they can be grouped together to form 9 classes, as in [9]. Each DT sequence includes 75 frames of 160×110 pixels. Here, the DT sequences are cropped to show the representative dynamics alone, thus, leading to frames of 48×48 pixels.

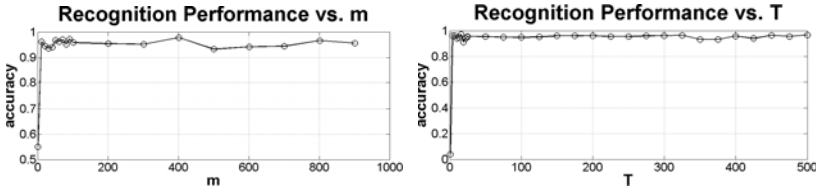
50-class breakdown: In the case of the 50 DT classes, the state-of-the-art recognition result (97.5%) was achieved by using kernel support vector machines (SVM's) [24]. Here, four cross-fold validation was performed, so the training set included $M = 150$ DT sequences (i.e. 3 sequences for each class). Applying *DL-PEGASOS* with $m = 150$ (i.e. $\Theta(M)$) and $T = 25$ iterations, we obtain an average recognition performance of 99% when both class dependent and class independent weights were learned. The class independent weights for the LBP, PHOG, and LDS descriptors are $w_1 = 1.95$, $w_2 = 1.12$, and $w_3 = 1.33$ respectively. This clearly indicates that the discrimination between DTs in this dataset is dominated by their spatial texture features, whereby using these features alone leads to a recognition rate of 90%. This reinforces the conclusion of [7], whose authors also reported on the dominant discriminative power of static texture in the UCLA DT dataset. In what follows, we will evaluate *DL-PEGASOS* on the 9-class breakdown of this dataset, since it poses a greater challenge.

9-class breakdown: In the case of the 9 DT classes, the state-of-the-art recognition result (80%) was achieved by using a bag-of-words model on LDS features [9][5], which lends itself useful to view-invariant recognition. For comparison, we adopt the same experimental setup as in [9]. We train on 50% of the dataset (i.e. $M = 100$) and test on the rest, with the recognition rates recorded as the average rate over 20 trials (i.e. random bisection of the classes in the dataset). First, we study the effect of the *DL-PEGASOS* free parameters (i.e. m and T) on the average recognition performance. Fig. 2(a) plots the recognition rate of class independent *DL-PEGASOS* when m is varied, while T is fixed to 25 iterations. Since $k = 1$, the total number of distance constraints is about 7000, from

[4] All experiments were executed using MATLAB 7.6 on a 2.4 GHz, 4GB RAM PC. Some *DL-PEGASOS* parameters were kept constant: **(i)** $k = 1$ nearest neighbors for $\mathcal{R}(.)$ and **(ii)** $\lambda = 0.05$.

[5] In [9], only 8 classes were considered, since the "plants" class was removed.

which m distance constraints are randomly chosen at each iteration. It is evident that recognition rate very quickly stabilizes ($\sim 95\%$), thus, indicating that most distance constraints do not play a significant role in discriminating between DT classes. This seems intuitive, since most constraints are easily satisfied for DTs that are significantly different in DT feature space. We also conclude that m can be reduced to $\Theta(M)$, without loss of performance. Similarly, Fig. 2(b) plots the recognition rate as T is increased, while m is fixed to 100. Clearly, the stable rate ($\sim 95\%$) is reached in a very small number of iterations.



(a) (b)

Fig. 2. Plots the recognition performance of *DL-PEGASOS* versus m (the number of distance constraints per iteration) and T (the maximum number of iterations) when class dependent weights are learned. To obtain the recognition rates in 2(a), we use $T = 25$. To obtain the recognition rates in 2(b), we use $m = 100$.

By setting $m = 100$ and $T = 25$, we obtain an average recognition rate of 95.6%, which significantly outperforms the state-of-the-art (80%) on this dataset. Fig. 3 shows the average confusion matrix for this experiment. The confused classes tend to have very similar appearance and/or dynamics, especially "fire" + "smoke", "flowers" + "plants" and "fountains" + "waterfall". In regards to time complexity, each complete trial ran in under 0.6 seconds. This time does not include feature extraction or pairwise elementary distance computation.

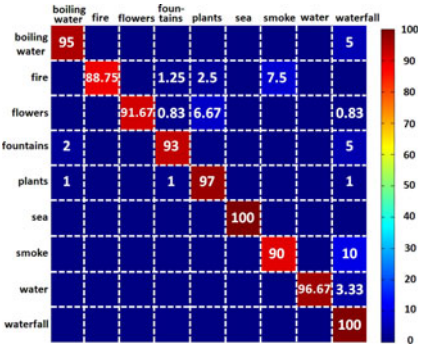


Fig. 3. Shows the confusion matrix for the 9-class experiment

Here, we mention that the recognition performance of class dependent *DL-PEGASOS* (82%) is significantly less than the class independent performance above. This is indicative of overfitting due to the small number of DTs per class. However, it is worthwhile to examine the values of $\boldsymbol{w}_c$, since they shed light on which DT dimension(s) are the most discriminative for a given class. From the weights in Table 1, we notice that some of our intuitions about what discriminates certain DTs are validated. For example, classes defined primarily by their spatial texture appearance (e.g. "flowers", "plants", and "sea") have

Table 1. Class dependent weights for the 9-class recognition experiment

	boiling water	fire	flowers	fountains	plants	sea	smoke	water	waterfall
w_1 (LBP)	0.21	1.22	**10.58**	0.12	**2.95**	**6.27**	4.23	7.13	4.73
w_2 (PHOG)	**7.81**	0.17	1.06	0.83	0.19	2.95	1.99	1.61	0.93
w_3 (LDS)	7.31	**7.07**	1.45	**10.18**	0.14	1.08	5.93	4.70	7.12

dominant w_1 values. Other classes that are primarily defined by their motion have dominant w_3 values (e.g. "fire" and "fountains"). Interestingly, the "boiling water" class is the only class where w_2 is the largest weight. This is due, in part, because the spatial texture is irregular and highly varying over time, while the overall layout remains stable. The other classes rely on a combination of these dimensions for their discriminative power.

5.2 DynTex++ Dataset

As mentioned before, the UCLA dataset is currently the benchmark for DT recognition, even though a much larger and more diverse datasets (the DynTex dataset [25]) exists. The UCLA dataset remains the benchmark due to the following reasons. **(i)** Its DT sequences have already been pre-processed from their raw form, whereby each sequence is cropped to show its representative dynamics in absence of any static or dynamic background. **(ii)** Only a single DT is present in each DT sequence. **(iii)** In each DT sequence, no panning or zooming is performed. **(iv)** Ground truth labels of the DT sequences are provided. Although some researchers have applied their recognition algorithms on the DynTex dataset (e.g. [20]), it is difficult to manage/use because it lacks the above four properties, in its present form. Therefore, we propose the compilation of a new dataset, called DynTex++.

Compiling the DynTex++ Dataset: The goal here is to organize the raw data in the DynTex dataset in order to provide a richer benchmark that is publicly available (http://vision.ai.uiuc.edu/∼bghanem2/DynTex++.htm) for future DT analysis, in the same way the UCLA dataset is currently. The original dataset is already publicly available ($\sim 2GB$ of data); however, only the raw AVI videos are provided. We proceeded to filter, pre-process, and label these DT sequences. While DynTex contains a total of 656 video sequences, DynTex++ uses only 345 of them. We eliminated sequences that contained more than one DT, contained dynamic background, included panning/zooming, or did not depict much motion. The remaining sequences were then hand labeled as one of $N = 36$ classes (e.g. "flying birds", "waterfall", "vehicle traffic"). They were not uniformly distributed among the N classes. We preprocessed them so each class contained the same number of subsequences.

The preprocessing proceeded as follows: **(i)** Each sequence is spatially downsampled by a factor of 0.75 and converted to grayscale. **(ii)** Since it is infeasible to manually crop these sequences, we randomly selected a large (1000) set of subsequences of fixed size ($50 \times 50 \times 50$), each of which is attributed a relevance score

that represents how much motion it entails. This score is the average optical flow [26] energy in the subsequence. By doing this, static background subsequences are eliminated from consideration and the more relevant DT subsequences remain. **(iii)** From each class, we selected 100 subsequences with the highest scores (uniformly chosen from the sequences constituting this class), thus, resulting in a dataset of $M = 3600$ subsequences. For more details on DynTex++, refer to the **supplementary material**.

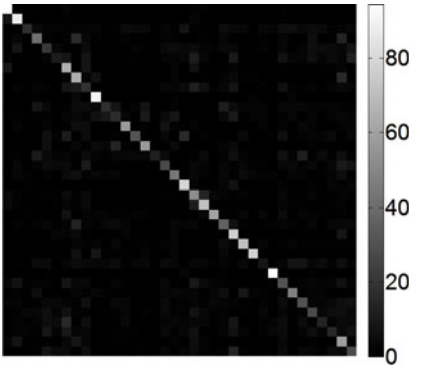


Fig. 4. shows the confusion matrix for DT recognition on DynTex++

***DL-PEGASOS* on DynTex++:** We apply our approach to the DynTex++ dataset, using an experimental setup similar to the one in the 9-class experiment on the UCLA dataset. In this case, we set $m = 2000$ and $T = 100$. We obtain an average recognition rate of 63.7%, with the average confusion matrix shown in Fig. 4. Each trial took under 15 seconds to run to completion.

6 Conclusions and Acknowledgments

In this paper, we formulate DT recognition in a maximum margin distance learning framework, where the distance between two DTs is a linear combination of three elementary distances representing DT space. These distance weights are efficiently learned by our proposed *DL-PEGASOS* algorithm, whose computational complexity is linear in the number of training samples. We validated our approach by outperforming the state-of-the-art on the UCLA benchmark, as well as, applying it the newly compiled DynTex++ dataset. 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. Shalev-Shwartz, S., Singer, Y., Srebro, N.: Pegasos: Primal estimated sub-gradient solver for svm. In: ICML (2007)
2. Ghanem, B., Ahuja, N.: Extracting a fluid dynamic texture and the background from video. In: CVPR (2008)
3. Soatto, S., Doretto, G., Wu, Y.N.: Dynamic textures. IJCV 51, 91–109 (2003)
4. Chetverikov, D., Peteri, R.: A brief survey of dynamic texture description and recognition. In: International Conference on Computer Recognition Systems (2005)
5. Saisan, P., Doretto, G., Wu, Y.N., Soatto, S.: Dynamic texture recognition. In: CVPR, pp. 58–63 (2001)
6. Martin, R.J.: A metric for arma processes. IEEE Trans. on Signal Processing 48, 1164–1170 (2000)

7. Chan, A.B., Vasconcelos, N.: Probabilistic kernels for the classification of auto-regressive visual processes. CVPR 1, 846–851 (2005)
8. Woolfe, F., Fitzgibbon, A.W.: Shift-invariant dynamic texture recognition. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 549–562. Springer, Heidelberg (2006)
9. Ravichandran, A., Chaudhry, R., Vidal, R.: View-invariant dynamic texture recognition using a bag of dynamical systems. In: CVPR, pp. 1651–1657 (2009)
10. Peteri, R., Chetverikov, D.: Dynamic texture recognition using normal flow and texture regularity. In: Iberian Conference on Pattern Recognition and Image Analysis (2005)
11. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray scale and rotation invariant texture analysis with local binary patterns. TPAMI 24, 971–987 (2002)
12. Zhao, G., Pietikainen, M.: Local binary pattern descriptors for dynamic texture recognition. In: ICPR, vol. 2, pp. 211–214 (2006)
13. Zhao, G., Pietikainen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. TPAMI 29, 915–928 (2007)
14. Frome, A., Singer, Y., Sha, F., Malik, J.: Image retrieval and classification using local distance functions. In: NIPS (2006)
15. Frome, A., Singer, Y., Sha, F., Malik, J.: Learning globally-consistent local distance functions for shape-based image retrieval and classification. In: ICCV (2007)
16. Gu, C., Lim, J.J., Arbelaez, P., Malik, J.: Recognition using regions. In: CVPR, pp. 1030–1037. IEEE, Los Alamitos (2009)
17. Varma, M., Ray, D.: Learning the discriminative power-invariance trade-off. In: ICCV (2007)
18. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: ICCV (2009)
19. Rubner, Y., Tomasi, C., Guibas, L.J.: A metric for distributions with applications to image databases. In: ICCV (1998)
20. Zhao, G., Pietikainen, M.: Dynamic texture recognition using volume local binary patterns. In: ECCV, Workshop on Dynamical Vision, pp. 12–23 (2006)
21. Wang, X., Han, T.X., Yan, S.: An hog-lbp human detector with partial occlusion handling. In: ICCV (2009)
22. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)
23. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: CIVR (2007)
24. Chan, A.B., Vasconcelos, N.: Classifying video with kernel dynamic textures. In: CVPR, vol. 1 (2007)
25. Peteri, R., Huiskes, M., Fazekas, S.: DynTex: the Centre for Mathematics and Computer Science (CWI), Amsterdam (2006), http://www.cwi.nl/projects/dyntex/
26. Gautama, T., Hulle, M.A.V.: A phase-based approach to the estimation of the optical flow field using spatial filtering. TNN 13, 1127–1136 (2002)

Image Invariants for Smooth Reflective Surfaces

Aswin C. Sankaranarayanan[1], Ashok Veeraraghavan[2],
Oncel Tuzel[2], and Amit Agrawal[2]

[1] Rice University, Houston, TX 77005, USA
[2] Mitsubishi Electric Research Labs, Cambridge, MA 02139, USA

Abstract. Image invariants are those properties of the images of an object that remain unchanged with changes in camera parameters, illumination etc. In this paper, we derive an image invariant for smooth surfaces with mirror-like reflectance. Since, such surfaces do not have an appearance of their own but rather distort the appearance of the surrounding environment, the applicability of geometric invariants is limited. We show that for such smooth mirror-like surfaces, the image gradients exhibit degeneracy at the surface points that are parabolic. We leverage this result in order to derive a photometric invariant that is associated with parabolic curvature points. Further, we show that these invariant curves can be effectively extracted from just a few images of the object in uncontrolled, uncalibrated environments without the need for any a priori information about the surface shape. Since these parabolic curves are a geometric property of the surface, they can then be used as features for a variety of machine vision tasks. This is especially powerful, since there are very few vision algorithms that can handle such mirror-like surfaces. We show the potential of the proposed invariant using experiments on two related applications - object recognition and pose estimation for smooth mirror surfaces.

1 Introduction

Image invariants are those properties of the images of an object that remain unchanged with changes in camera parameters, illumination etc. Any geometric invariant (eg., cross ratio) is true for surfaces with any reflectance characteristics including diffuse, specular and transparent surfaces. But, in order to actually use these geometric invariants from observed images of an object, one needs to identify point correspondences across these images. Establishing such point correspondences from images of diffuse objects is a meaningful task since these objects have photometric features of their own. But surfaces with mirror reflectance do not have an appearance of their own, but rather present a distorted view of the surrounding environment. Therefore, establishing physical point correspondences using image feature descriptors (such as SIFT) is not meaningful. Such descriptors find correspondences between environment reflections, and therefore are not physically at the same point on the surface. Thus, there is a need to find photometric properties of specular surfaces that are invariant to the surrounding environment. In this paper, we study and present such an invariant for the images of smooth mirrors.

The main results of this paper arise by studying the photometric properties of the images of mirror surfaces around points that exhibit parabolic curvature. Parabolic curvature points are fundamental to perception of shape both for diffuse [15, 17] and for

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 237–250, 2010.

specular surfaces [23]. In this paper, we first derive a photometric invariant that is associated with parabolic curvature points of the mirror surface. We show that a smooth mirror imaged by an orthographic camera, reflecting an environment feature at infinity, exhibits degenerate gradients at parabolic curvature points. This degeneracy is characterized by the image gradients being orthogonal to the direction of zero curvature at the parabolic point. Although the invariant holds exactly for the aforementioned imaging setup, we empirically show that for a range of practical imaging conditions (with perspective camera and finite scene), the invariance still holds to a high degree of fidelity.

The set of parabolic points is a geometric property of a surface and each surface has its own distinct set of parabolic curves. The photometric invariant that we propose allows us to detect these parabolic curves from just images of the specular object without any a priori knowledge about its 3D shape or the surrounding environment. Since these parabolic points are a geometric property of the surface, they can then be used for a variety of machine vision tasks such as object recognition, pose estimation and shape regularization. In this paper, we demonstrate a few such applications.

Contributions: The specific technical contributions of this paper are:

- We present a theoretical study of the properties of images of mirrors. We show that under a certain imaging setup, the image derivatives at the points of parabolic curvature exhibit degeneracies independent of the surrounding environment.
- We show that this degeneracy can be measured quantitatively using just a few images of the object under arbitrary illumination, thereby allowing us to recover the parabolic curvature points associated with the mirror.
- We show new applications of this invariant to challenging machine vision problems such as object recognition and pose estimation for mirror objects.

2 Prior Work

In this paper, we are interested in identifying invariants for images of mirrors. Additional assumptions are needed for obtaining something meaningful/non-trivial. A planar mirror viewed by a perspective camera is optically the same as a perspective camera, and hence, can produce arbitrary images.

The qualitative properties of images of specular/mirror objects have been well studied (see [14] for a survey). Zisserman et al. [25] show that local surface properties such as concave/convexity can be determined under motion of the observer without knowledge of the lighting. Blake [4] analyze stereoscopic images of specular highlights and show that disparity of highlights observed on the mirror is related to the qualitative properties of the shape such as its convexity/concavity. Blake and Brelstaff [5] quantify local surface ambiguities given stereo images of highlights. Fleming et al. [11] discuss human perception of shape from images of specular objects even when the environment is unknown and show that humans are capable of accurately determining the shape of the mirror; potentially from image compression cues. In another study of human perception of specular surfaces, Savarese et al. [20] report poor perception when the surrounding scene comprises of unknown but structured patterns.

It is worth noting that points/curves of parabolic curvature have been studied in terms of their photometric properties. Our search is motivated in part from classical results in photometric stereo and more recent work in the area of specular flow. Koenderink and van Doorn [15] demonstrate that the structure of isophotes is completely determined by parabolic curves of the surface. Further, they also show that the a local extremum of the field of isophotes occur on parabolic curves, and move along these curves under motion of the light source. Isophotes as a construct are useful for diffuse objects with constant albedo and mirrors under simple lighting (such as a point light source), and do not extend well to scenes/object with rich textures.

Much of prior work using properties of parabolic points revolve around the idea of consistency of highlights at parabolic curvature points across small changes in view or illumination. Miyazaki et al [18] use parabolic curves for registration of transparent objects across views.

Recent literature has focused on estimating the shape of the mirror from the specular flow [1,21,7] induced under motion. Specular flow is defined as the movement of environmental features on the image of a mirror due to motion of the mirror/scene. It has been shown that parabolic curvature points exhibit infinite flow under infinitesimal motion. The infinite flow is a result of appearance of new scene features and disappearance of existing ones, an observation made earlier by Longuet-Higgins [16] and Walden and Dyer [22] as well. Waldon and Dyer [22] suggest that, for mirrors, reliable qualitative shape information is available only at the parabolic curves in the forms of discontinuous image flow fields. Studies in perception [17] hint at the ability of humans to detect and use parabolic curvature curves to perform local shape analysis. Some existing approaches in perception [23] and detection [9] of mirrors remark on the anisotropy of gradients in the images of smooth mirrors. However, these papers do not identify the existence of the invariance, the assumptions required for the invariance to hold, the geometric interpretation behind its occurrence and the stability of the invariance for practical imaging scenarios. More importantly, in addition to exploring these properties, we also show that parabolic curvature points of the mirror (a surface descriptor) can be recovered from a few images of the mirror.

3 Deriving the Invariant

This section describes the main technical contributions of the paper. We begin with a brief overview of parabolic curvature points. Then, we discuss the image formation model for mirror objects and show that the observed image gradients exhibit a degeneracy at the points of parabolic curvature, irrespective of the environment. This leads us to define an invariant for the surfaces of smooth mirrors.

3.1 Parabolic Curvature Points

Let us model the shape of the (smooth) mirror in its Monge form $(x, y, f(x, y)) = (\mathbf{x}, f(\mathbf{x}))$ in a camera coordinate system where the function f is twice continuously differentiable. At a given point on the surface, the curvature along a curve is defined as the reciprocal of the radius of the osculating circle. The principal curvatures are

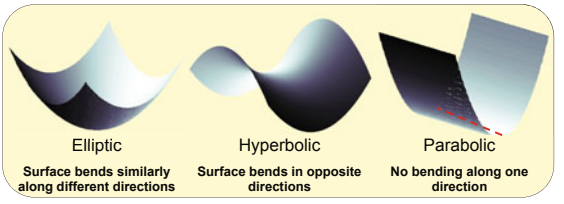


Fig. 1. Local properties of a surface can be classified into 4 types: elliptic, hyperbolic, parabolic and flat umbilic (planar). This classification deals with the bending of the local surface in various directions. The parabolic curve is shown in red.

defined as the minimum and maximum values of the curvature measured along various directions at a given point. The product of the principal curvatures is defined as the Gaussian curvature. It can be shown [15] that the Gaussian curvature is given by

$$\frac{f_{xx}f_{yy} - f_{xy}^2}{1 + f_x^2 + f_y^2}. \tag{1}$$

Points at which one of the principal curvatures is zero are termed parabolic curvature points or simply parabolic points. Defining the Hessian at point $\mathbf{x}$ of the surface as

$$H(\mathbf{x}) = \frac{1}{2}\begin{bmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{bmatrix}_{(\mathbf{x})}, \tag{2}$$

parabolic curvature points are defined by points where $\mathrm{rank}[H(\mathbf{x})] = 1$. When both principal curvatures at a point are zero, the point is referred to as *flat umbilic*. Planes are examples of surfaces which are flat umbilic everywhere. Shown in Figure 1 are characterization of local properties of a surface.

3.2 Image Formation for Mirror Objects

Mirrors do not have an appearance of their own, and image of mirror are warps of the surrounding environment. Modeling the shape of the mirror as $(\mathbf{x}, f(\mathbf{x}))$, image formation can be described by identifying the camera and the environment. We model the camera as orthographic. Under an orthographic camera model, all the rays entering the camera are parallel to its principal direction.

The surface gradient at pixel location $\mathbf{x}$ is given as $\nabla f = (f_x, f_y)^T$, and the surface normal is given as

$$\mathbf{n}(\mathbf{x}) = \frac{1}{\sqrt{1 + \|\nabla f\|^2}} \begin{pmatrix} -\nabla f \\ 1 \end{pmatrix}. \tag{3}$$

The camera viewing direction $\mathbf{v}$ at each pixel is the same, $\mathbf{v} = (0, 0, 1)^T$. Under perfect mirror reflectance, we can compute the direction of the ray that is reflected onto the camera as $\mathbf{s} = 2(\mathbf{n}^T\mathbf{v})\mathbf{n} - \mathbf{v}$ The corresponding Euler angles $\Theta(\mathbf{x}) = (\theta, \phi)$ are given as,

$$\tan\phi(\mathbf{x}) = \frac{f_y}{f_x}, \qquad \tan\theta(\mathbf{x}) = \frac{2\|\nabla f\|}{1 - \|\nabla f\|^2}. \tag{4}$$

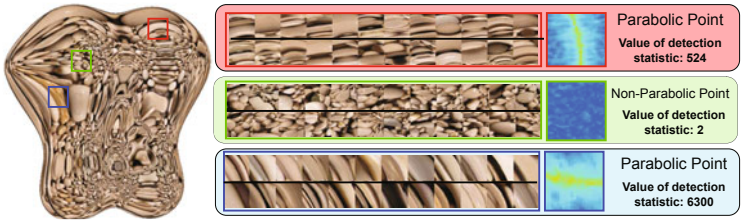


Fig. 2. Degeneracy of image gradients at parabolic points. Shown are image patches at three locations on a mirror from *multiple* images rendered under a rotating environment. The image patches corresponding to parabolic curves do not have have gradients along the parabolic curve. This is in contrast to a non-parabolic point which can have arbitrary appearance.

The scene/environment that is seen at pixel $\mathbf{x}$ is hence defined by the intersection of the scene and the ray in the direction of $\mathbf{s}(\mathbf{x})$ from the location of the mirror element $(\mathbf{x}, f(\mathbf{x}))^T$. In the special case of environment at infinity, the dependence on the location of the mirror is completely suppressed, and the environment feature observed depends only on the surface gradient ∇f.

Under the assumption of environment at infinity, we can define the environment map over a sphere. Let $E : \mathbb{S}^2 \mapsto \mathbb{R}$ be the environment map defined on the sphere $\mathbb{S}^2$ under the Euler angle parametrization. Under *no inter-reflectance* within the object, the forward imaging equation for the intensity $I(\mathbf{x})$ observed at pixel $\mathbf{x}$ is given as

$$I(\mathbf{x}) = E(\Theta(\mathbf{x})) \tag{5}$$

where $\Theta(\mathbf{x})$ is the Euler angle of the observed ray as given in (4). Differentiating (5) with respect to $\mathbf{x}$, the image gradients are given by

$$\nabla_{\mathbf{x}} I = 2H(\mathbf{x}) \left[\frac{\partial \Theta}{\partial \nabla f}\right]^T \nabla_{\Theta} E \tag{6}$$

where the Hessian $H(\mathbf{x})$ is defined in (2). The full derivation is in the supplemental material (and similar to that of [1]).

For parabolic curvature points, $H(\mathbf{x})$ is singular. As a consequence, $\nabla_{\mathbf{x}} I$ takes values that are proportional to the non-zero eigenvalue of $H(\mathbf{x})$, immaterial of what the environment gradient $\nabla_{\Theta} E$ is. Figure 2 shows the local appearance of parabolic and non-parabolic points under various environment maps.

3.3 Invariant

Given a smooth mirror $(\mathbf{x}, f(\mathbf{x}))$, where f is $\mathbb{C}^2$ continuous, placed with the surrounding environment at infinity and viewed by an orthographic camera, the proposed invariant is a statement on the observed image gradient at parabolic curvature points of the mirror. Under this setting, the image gradients at parabolic curvature points are *degenerate* and take values along a single direction that is defined by the local shape of the surface. This property is independent of the scene in which the mirror is placed.

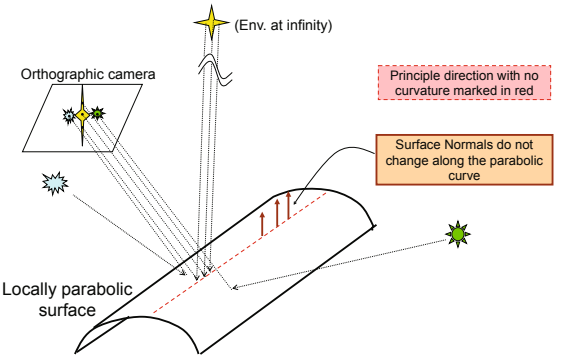


Fig. 3. The proposed invariant can be geometrically described using the ray diagram above. At a parabolic point, by definition, the normal (and curvature) do not change along one direction. Under orthographic imaging and scene at infinity, the same feature is imaged onto the camera as we move along the parabolic curve and hence, the image gradient disappears along this direction.

The invariant arises directly due to the principal direction of zero curvature at parabolic point (see Figure 3). By definition, an infinitesimal movement on the surface along this direction does not change the surface normal. Under our imaging model, the environment feature imaged at a point depends only on the surface normal. Hence, an infinitesimal displacement on the image plane along the projection of this direction does not change the environment feature imaged. As a consequence, the image gradient along this direction is zero. This geometric understanding is related to recent work on ray-space specular surface analysis [24,10], where the authors study local mirror patches as general linear cameras and associate different camera models to different local surface properties. Interesting connections to our work can potentially be derived from these papers and remain an important direction for future research.

Mathematically, the invariant can be expressed in various forms. From (6) and under the assumed imaging conditions, a parabolic curvature point at $\mathbf{x}_0$ satisfies

$$\nabla_{\mathbf{x}} I(\mathbf{x}_0) = \|\nabla_{\mathbf{x}} I(\mathbf{x}_0)\| \mathbf{v} \tag{7}$$

where $\mathbf{v}$ is the eigenvector of $H(\mathbf{x}_0)$ with non-zero eigenvalue. An alternate interpretation that does not involve $H(\mathbf{x})$ uses the matrix $M(\mathbf{x})$ defined as:

$$M(\mathbf{x}) = \sum_E \left((\nabla_{\mathbf{x}} I(\mathbf{x}; E))(\nabla_{\mathbf{x}} I(\mathbf{x}; E))^T \right) \tag{8}$$

where $I(x; E)$ is the intensity observed at pixel $\mathbf{x}$ under environment defined in $E(\Theta)$. Note that the summation in (8) is over all possible environment maps. At points of parabolic curvature,

$$\text{rank}[M(\mathbf{x}_0)] = 1 \tag{9}$$

In contrast, for elliptic and hyperbolic points, the matrix M is full rank. For flat umbilical points, $H(\mathbf{x})$ is the zero matrix and the image gradients are zero as well. Therefore, the matrix defined in (8) will be zero rank.

4 Detecting Parabolic Curvature Points

We derive a practical algorithm for detecting points of parabolic curvature from multiple images of a mirror. The algorithm exploits the consistency (or degeneracy) of the image gradients as given in (9). Under motion of the camera-mirror pair (or equivalently, rotation of the environment), the environment feature associated with each point changes arbitrarily. However, parabolic points are *tied* to the surface of the mirror, and hence, the direction of image gradients associated with them do not change. This motivates an acquisition setup wherein the environment is changed arbitrary and consistency of image gradient at a pixel indicates whether or not it has parabolic curvature. Since movement of the camera-object pair simultaneously is the equivalent to that of rotation of the environment, we use environment rotation to denote both. In practise, moving the camera-object pair is easier to accomplish.

Given a set of images $\{I_j\}$, compute the matrix

$$M(\mathbf{x}) = \sum_j \left(\nabla_{\mathbf{x}} I_j(\mathbf{x})\right) \left(\nabla_{\mathbf{x}} I_j(\mathbf{x})\right)^T \tag{10}$$

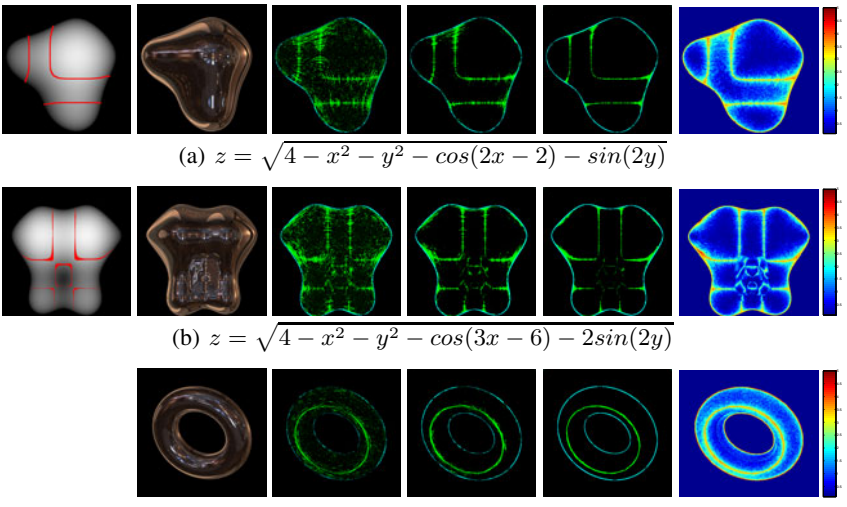
using image gradients computed at each frame. We use the ratio of the eigenvalues of $M(\mathbf{x})$ as the statistic to decide whether or not a pixel $\mathbf{x}$ observes a parabolic curvature point. Figure 4 shows estimates of parabolic points of different surfaces. Images for this experiment were rendered using PovRay. Each image was taken under a arbitrary rotation of the environment. As the number of images increase, the detection accuracy increases significantly as $M(\mathbf{x})$ at non-parabolic points become well-conditioned.

We believe that our approach is unique in the sense that it recovers a 'dense' characterization of parabolic curvature points from uncalibrated images of a mirror. Much of the existing literature on using the photometric properties of parabolic points rely on the stability of highlights at parabolic points under changes in views. However, such a property is opportunistic at best, and does not help in identifying all the parabolic curvature points associated with the visible surface of the mirror. In this sense, the ability to recover a dense set of parabolic curvature points opens the possibility of a range of applications. We discuss these in Section 5.

Theoretically, the invariance is guaranteed only for an orthographic camera and an environment at infinity. However, in practice, the invariance holds with sufficient fidelity when these assumptions are relaxed. We explore the efficacy of the proposed invariant for a range of practical operating conditions in Section 6.

Mis-detection: The proposed invariant does not take inter-reflections into account. Inter-reflections alter the physics of the imaging process locally, and violates the relations made earlier in physical models. Imaging resolution also affects the detection process. For low resolution images, the curvature of the surface observed in a single pixel might deviate significantly from parabolic. Such a scenario can potentially annul the invariance at the parabolic point due to corruption from the surrounding regions.

False Alarm: It is noteworthy that the invariant describe image gradients at parabolic points. However, degenerate gradients do not necessarily imply the presence of parabolic points. Clearly, for small number of images, it is possible that a surface pixel/



(a) $z = \sqrt{4 - x^2 - y^2 - cos(2x - 2) - sin(2y)}$

(b) $z = \sqrt{4 - x^2 - y^2 - cos(3x - 6) - 2sin(2y)}$

(c) Rotated Torus

Fig. 4. Detecting parabolic curvature points from rendered images for various surfaces. (From left to right) the depth map of the mirror with the parabolic points highlighted in red; a rendered image of the surface using the *Grace cathedral* environment map; detected parabolic points from 2 images; from 5 images; from 25 images; Log of decision statistic estimated from 25 images. The occluding contour is shown in cyan and the parabolic points in green.

patch do not observe environment features that are sufficiently rich. Similarly, discontinuities in the surface such as occluding contours can lead to consistent degeneracy in the observed image gradients.

Note that, the detection of statistics does not require the environment texture to be rich. Using increasing number of images (camera-mirror pair rotations), the degeneracies due to environment become incoherent and can be filtered out easily. Our experiments include textures such as the Grace cathedral which exhibit large regions with little or no textures and the method succeeds to capture the statistics regardless.

5 Applications

In this section, we describe three applications of the presented theory; (1) pose estimation, (2) recognition of mirror-like objects; and (3) a possible extension for surface reconstruction. The equivalent algorithms designed for diffuse/textured surfaces require establishing correspondences between image observations and a model of the object [13]. For specular objects, the highlights on the objects serve as an informative cue for object detection and pose estimation [8]. Similarly, Gremban and Ikeuchi [12] use specular highlights for object recognition, and plan novel views that are discriminative between objects with similar highlights. However, these methods do not generalize to objects with mirror reflectance. In a calibrated setup (camera and environment), it is possible to infer about surface normals through image and environment

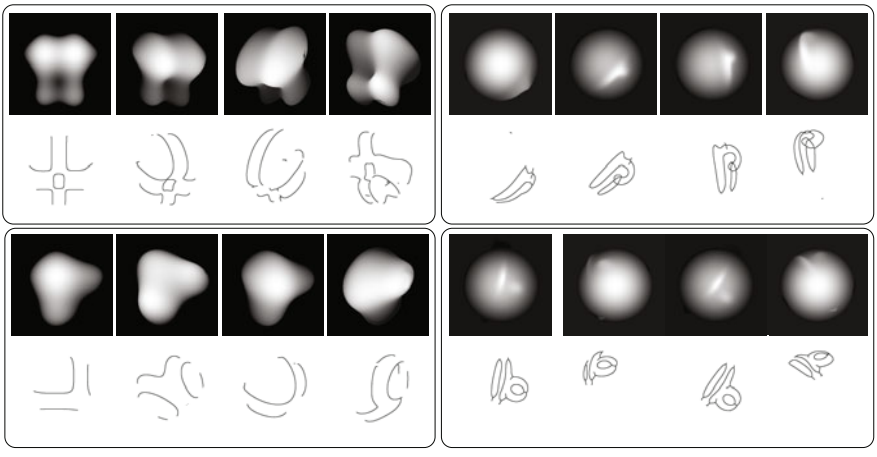


Fig. 5. Parabolic curves provide a unique signature for object pose and identity. Shown are the parabolic curves of four different objects in differnet poses. In each instance, the depth map and the analytically computed parabolic curves are shown.

correspondences which can be further utilized for estimation algorithms [6,19]. However the required calibration process is a tedious task. Pose estimation and recognition in an uncalibrated setup remains to be a challenge and we show that the proposed image invariants provide necessary information for such tasks.

Pose Estimation: The pose estimation algorithm recovers the 3D rotation and 2D translation parameters with respect to a nominal pose of the object. Since the camera model is assumed orthographic, the object pose can only be recovered up to depth ambiguity. We assume that either the parametric form or the 3D model of the object is given in advance. Based on the representation, the 3D positions of the parabolic points at object coordinates are recovered either analytically (using parametric form) or numerically (using the 3D model). In an offline process, we generate a database of curve templates by rotating the parabolic curvature points with respect to a set of sampled 3D rotations and projecting visible points to the image plane. Since rotation of the object along the principal axes (θ_z) of the camera results in an in-plane rotation of the parabolic points on the image plane, it suffices to include only out-of-plane rotations (θ_x and θ_y) to the database which is performed by uniform sampling of the angles on the 2-sphere. A few samples included into the database is given in Figure 5.

The initial pose of the object is recovered by searching for the database template together with its optimal 2D Euclidean transformation parameters $\mathbf{s} = (\theta_z,\, t_x,\, t_y)$, which aligns the parabolic points of the template to the image parabolic curvature points. We use a variant of chamfer matching technique [2] which measures the similarity of two contours. The precision of the initial pose estimation is limited by the discrete set of out-of-plane rotations included into the database. We refine the estimation using a combination of iterative closest point (ICP) [3] and Gauss-Newton optimization algorithms.

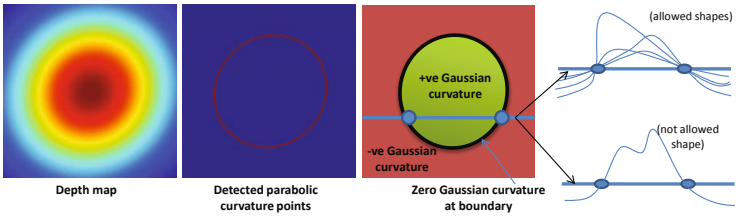


Fig. 6. A stylistic example showing how the parabolic points provide global shape priors. Such priors are extremely useful in restricting the solution space in a surface recovery algorithm as well as in identifying regions where simple parametric models describe the surface accurately.

Recognition: The parabolic curvature points provide unique signatures to recognize many objects in variable poses. The object recognition algorithm is a simple extension of the presented pose estimation approach. For each object class we repeat the pose estimation process and recover the best pose parameters. The object class is then given by the minimum of the chamfer cost function [2] over all classes.

Shape Priors: Knowledge of the parabolic curvature points gives a strong prior on the shape of the mirror. It is well known that curves of parabolic curvature separate regions of elliptic and parabolic curvature. Toward this end, we can constrain the range of possible shapes (Figure 6). Further, in each region we can use simple non-parametric surface models such as splines and regularize their parameters to satisfy the curvature properties. This forms a compelling direction for future research.

6 Experiments

We use both real and synthetically generated images for our experiments. For synthetic experiments, we use publicly available ray-tracing software POV-Ray for photo realistic rendering which provides high quality simulations of real world environments including inter-reflections. Real data was collected with a Canon SLR camera using a 300mm lens, and placing the mirror approximately 150cm from the camera. Both camera and mirror were rigidly mounted to a platform, which was moved around to change the environment features seen on the mirror.

6.1 Detecting Parabolic Curves

In Figure 4, we present results for detection of parabolic curvature points from synthetically rendered images under the ideal imaging condition of orthographic camera and scene at infinity. We show the performance of the detection when these assumptions are violated. Figure 7 shows the detection of parabolic points when the scene is at a finite distance from the mirror. In particular, the detection of parabolic curvature points is reliable even when the minimum distance of the mirror to the object is the same as that of the variations in the depth of the object itself. This shows the stability of the detection statistic to finite scenes. Figure 8 shows stable detection results when the camera is

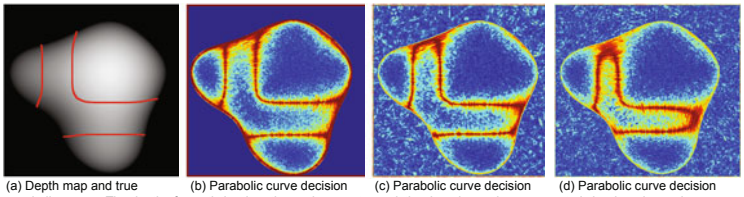


Fig. 7. Detection of parabolic points when the environment is at a finite distance from the mirror

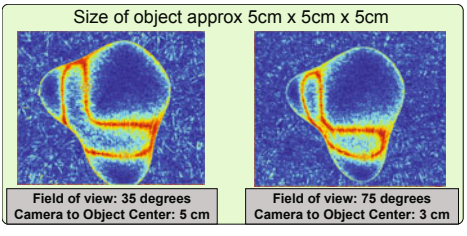


Fig. 8. Detection of parabolic points using a perspective camera under medium to large deviation from the orthographic case. The parabolic curvature points remain stable in both cases. Note that as the camera approaches the object and the field of view of the camera is increased, the relative locations of the (projection of the) parabolic points on the image plane changes. This, in part, explains the drift of the parabolic curvature points.

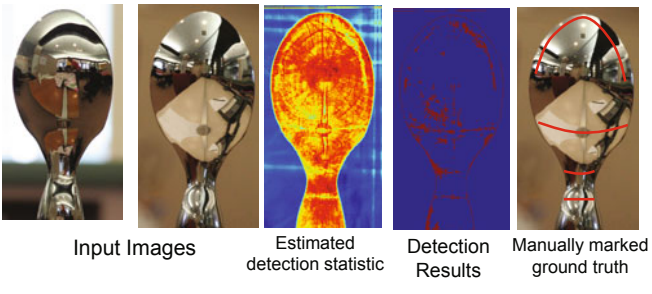


Fig. 9. Estimation of parabolic points of a real object using multiple images. Results were estimated using 17 images.

heavily perspective. These figures reinforce the detection of parabolic curves based on the invariant for practical imaging scenarios. In Figures 9 and 10, we show detection of parabolic curvature points using real images for two highly reflective objects.

6.2 Pose Estimation and Recognition

In the synthetic experiments, we randomly sample six parameters of the 3D object pose and render the object under several environment rotations. The parabolic curves on the

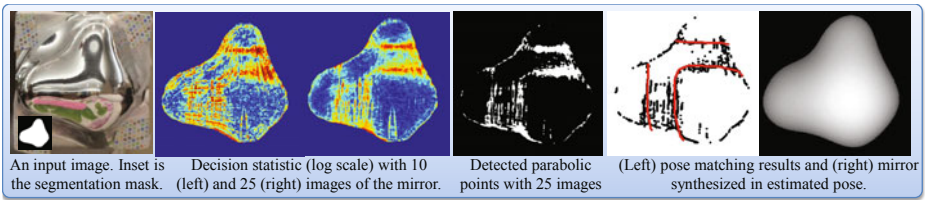
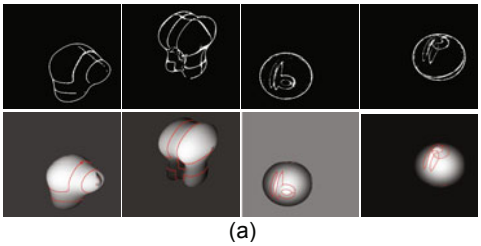


Fig. 10. Estimation of parabolic points of a real surface using variable number of images. The parametric form of the surface was given in Figure 4 and it is manufactured using a CNC machine. As the number of images increase, the degeneracies due to environment become incoherent and detection becomes more reliable.



(a)

Mean Abs. Err.	t_x	t_y	θ_x	θ_y	θ_z
Surface-1	1.94	1.60	0.55	0.67	0.46
Surface-2	1.86	2.87	0.96	0.59	0.28
Surface-3	3.54	8.55	2.74	1.04	0.93
Surface-4	4.63	6.69	1.95	1.39	0.48

(b)

Fig. 11. (a) Visualization of the pose estimation results. For each test object, we show the pose estimate at one of the 30 random poses used. (Top) Parabolic curvature points detected from 25 images of the mirror under a rotating environment. (bottom) Estimated pose of the mirror with the true parabolic curvature points overlaid. (b) Mean pose estimation errors. Translational error is in pixels and rotational error is in degrees. The results are averaged over 30 trials.

image plane is detected using the 25 rendered images which are then utilized to recover the object pose via the algorithm described earlier.

We provide results for four different surfaces which are shown in Figure 5. In Figure 11a, we present several pose estimation results. The simulation is repeated 30 times for each surface using a different pose and mean absolute estimation errors for five parameters of the 3D pose is given in Figure 11b. We note that, since the camera model is orthographic, the object pose can be recovered only up to a depth ambiguity. In all our trials the pose estimation algorithm converged to the true pose. As shown, the parabolic curves provide extremely robust features for pose estimation, and average rotation error is less than 2 degrees and 5 pixels. In Figure 10, we show pose detection results on real images of a mirror. The estimated pose was $(-4.24, -1.66, 1.9)$ for a ground truth of $(0, 0, 0)$.

For recognition, we place four objects simultaneously to the environment and recognize identities of these surfaces. This is a challenging scenario due to heavy interreflections of the surfaces. The same rendering scenario of the pose estimation experiment is repeated. The object identities are given via the minimum of the cost function after pose estimation. The average recognition rate over 10×4 trials is 92.5% and typical recognition examples are shown in Figure 12. We note that, two of the surfaces have

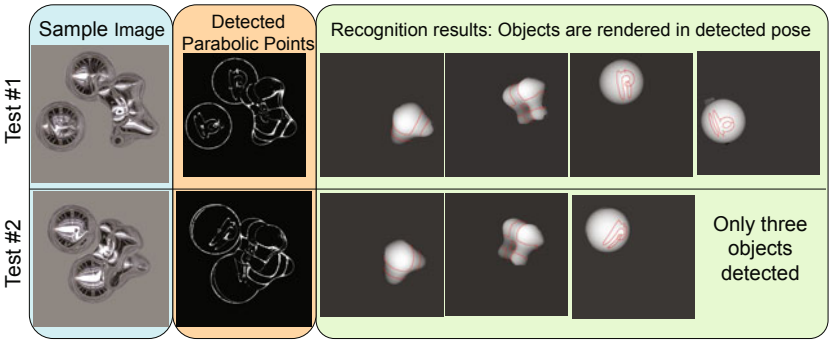


Fig. 12. Recognition experiment on synthetic images. Our test setup consisted of arbitrarily placing all four test objects in a virtual scene and rendering multiple images. The recovered parabolic curvature points were used to recognize the object and estimate its pose.

exactly the same occluding contour, therefore in this scenario this statistic is expected to fail whereas parabolic curvature points provide unique signatures.

7 Conclusions

In this paper, we propose a photometric invariant for images of smooth mirror. We show that images of mirror exhibit degenerate image gradients at parabolic curvature points when the camera is orthographic and the scene is at infinity. We demonstrate the practical effectiveness of the invariant even under deviations from this imaging setup. In particular, the invariant allows for a dense recovery of the point of parabolic curvature from multiple images of the mirror under motion of the environment. This allows us to recover a geometric property of the mirror. We show that recovery of the parabolic curvature points opens up a range of novel applications for mirrors.

Acknowledgments. We thank the anonymous reviewers for their feedback in improving the paper. We also thank Jay Thornton, Keisuke Kojima, John Barnwell, and Haruhisa Okuda, Mitsubishi Electric, Japan, for their help and support. Aswin Sankaranarayanan thanks Prof. Richard Baraniuk at Rice University for his encouragement and support.

References

1. Adato, Y., Vasilyev, Y., Ben-Shahar, O., Zickler, T.: Toward a Theory of Shape from Specular Flow. In: ICCV (October 2007)
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: Joint Conf. on Artificial Intelligence, pp. 659–663 (1977)
3. Besl, P., McKay, H.: A method for registration of 3-D shapes. TPAMI 14(2), 239–256 (1992)
4. Blake, A.: Specular stereo. In: Int. Joint Conf. on Artificial Intelligence, pp. 973–976 (1985)

5. Blake, A., Brelstaff, G.: Geometry from specularities. In: ICCV, pp. 394–403 (1988)
6. Bonfort, T., Sturm, P.: Voxel carving for specular surfaces. In: ICCV (October 2003)
7. Canas, G.D., Vasilyev, Y., Adato, Y., Zickler, T., Gortler, S., Ben-Shahar, O.: A Linear Formulation of Shape from Specular Flow. In: ICCV (September 2009)
8. Chang, J., Raskar, R., Agrawal, A.: 3D Pose Estimation and Segmentation using Specular Cues. In: CVPR (June 2009)
9. DelPozo, A., Savarese, S.: Detecting specular surfaces on natural images. In: CVPR (June 2007)
10. Ding, Y., Yu, J., Sturm, P.: Recovering specular surfaces using curved line images. In: CVPR (June 2009)
11. Fleming, R.W., Torralba, A., Adelson, E.H.: Specular reflections and the perception of shape. Journal of Vision 4(9), 798–820 (2004)
12. Gremban, K., Ikeuchi, K.: Planning multiple observations for object recognition. IJCV 12(2), 137–172 (1994)
13. Haralick, R., Joo, H., Lee, C., Zhuang, X., Vaidya, V., Kim, M.: Pose estimation from corresponding point data. IEEE Trans. on Systems, Man and Cybernetics 19(6), 1426–1446 (1989)
14. Ihrke, I., Kutulakos, K.N., Lensch, H.P.A., Magnor, M., Heidrich, W., Ihrke, I., Stich, T., Gottschlich, H., Magnor, M., Seidel, H.P.: State of the Art in Transparent and Specular Object Reconstruction. In: IEEE Int. Conf. on Image Analysis and Processing, vol. 12, pp. 188–193 (2005)
15. Koenderink, J., Van Doorn, A.: Photometric invariants related to solid shape. Journal of Modern Optics 27(7), 981–996 (1980)
16. Longuet-Higgins, M.S.: Reflection and refraction at a random moving surface. I. Pattern and paths of specular points. Journal of the Optical Society of America 50(9), 838 (1960)
17. Mamassian, P., Kersten, D., Knill, D.: Categorical local-shape perception. Perception 25, 95–108 (1996)
18. Miyazaki, D., Kagesawa, M., Ikeuchi, K.: Transparent surface modeling from a pair of polarization images. TPAMI 26(1), 73–82 (2004)
19. Savarese, S., Chen, M., Perona, P.: Local shape from mirror reflections. IJCV 64(1), 31–67 (2005)
20. Savarese, S., Fei-Fei, L., Perona, P.: What do reflections tell us about the shape of a mirror? In: Applied Perception in Graphics and Visualization (August 2004)
21. Vasilyev, Y., Adato, Y., Zickler, T., Ben-Shahar, O.: Dense specular shape from multiple specular flows. In: CVPR (June 2008)
22. Waldon, S., Dyer, C.: Dynamic shading, motion parallax and qualitative shape. In: IEEE Workshop on Qualitative Vision. pp. 61–70 (1993)
23. Weidenbacher, U., Bayerl, P., Neumann, H., Fleming, R.: Sketching shiny surfaces: 3D shape extraction and depiction of specular surfaces. ACM Transactions on Applied Perception 3(3), 285 (2006)
24. Yu, J., McMillan, L.: Modelling Reflections via Multiperspective Imaging. In: CVPR (June 2005)
25. Zisserman, A., Giblin, P., Blake, A.: The information available to a moving observer from specularities. Image and Vision Computing 7(1), 38–42 (1989)

Visibility Subspaces: Uncalibrated Photometric Stereo with Shadows

Kalyan Sunkavalli, Todd Zickler, and Hanspeter Pfister

Harvard University
33 Oxford St., Cambridge, MA, USA, 02138
{kalyans,zickler,pfister}@seas.harvard.edu

Abstract. Photometric stereo relies on inverting the image formation process, and doing this accurately requires reasoning about the visibility of light sources with respect to each image point. While simple heuristics for shadow detection suffice in some cases, they are susceptible to error. This paper presents an alternative approach for handling visibility in photometric stereo, one that is suitable for uncalibrated settings where the light directions are not known. A surface imaged under a finite set of light sources can be divided into regions having uniform visibility, and when the surface is Lambertian, these regions generally map to distinct three-dimensional illumination subspaces. We show that by identifying these subspaces, we can locate the regions and their visibilities, and in the process identify shadows. The result is an automatic method for uncalibrated Lambertian photometric stereo in the presence of shadows, both cast and attached.

1 Introduction

Photometric stereo seeks to recover the geometry of a scene by analyzing appearance changes under varying illumination. In spite of being based on a crude reflectance model, Lambertian photometric stereo is one approach that is frequent used. One of the reasons for the utility of Lambertian photometric stereo is its support of auto-calibration. In the ideal case, given a set of images under varying, but unknown, directional lighting, it is possible to recover both a surface normal field and the light source directions up to a three-parameter family of solutions [7,33].

Like any photometric stereo technique, uncalibrated Lambertian photometric stereo relies on inverting the image formation process. It seeks to explain observations using combinations of light sources, surface normals, and surface albedos; and in order to succeed, it must be able to reason effectively about which light sources are visible to each surface point. This problem is deceptively hard because shadowing is a non-local function of surface geometry, and heuristics for shadow detection, such as simple thresholding, are unreliable in the presence of albedo variations and sparse input images.

In this paper, we avoid explicit shadow detection by reasoning about illumination subspaces instead. It is well-known that the set of images of a convex

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 251–264, 2010.

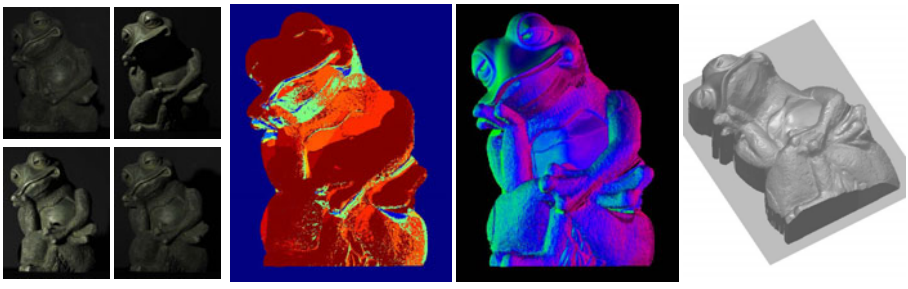


Fig. 1. Uncalibrated photometric stereo with shadows. From a sparse set of images of a Lambertian scene (a), we identify regions that can see a common set of lights (b) through subspace estimation. This provides per-pixel visibility and allows the recovery of surface normals (c) and light directions up to the standard global linear ambiguity. Integrating these normals produces a reconstruction (d) that is not corrupted by the strong shadowing in the input images.

Lambertian surface under directional lighting spans a three-dimensional linear subspace. It is also well-known that attached shadows and cast shadows violate this subspace property, so that the image-span of a scene with shadows can grow to a high dimension. What has not been fully exploited is that these high-dimensional spans have useful structure. We show that the image-span of any Lambertian scene captured under a discrete set of light sources with arbitrary shadowing can be decomposed into a set of three-dimensional subspaces. We refer to these as *visibility subspaces* because they correspond to sets of surface points that can see a common set of lights.

Given a sequence of uncalibrated photometric stereo images of a Lambertian object, the visibility subspaces can be automatically identified—without knowledge of the lighting directions—using well-known subspace clustering techniques. We show that once these subspaces are identified, the surface is partitioned, the exact set of lights that is visible to each region can be computed, and the surface and light directions can be reconstructed up to the usual global linear ambiguity.

2 Related Work

Photometric stereo can produce per-pixel estimates of surface normals and is a common technique for scene reconstruction. Originally developed for Lambertian surfaces and calibrated directional lighting [29], photometric stereo has been generalized to handle uncalibrated directional lights [15], specular and glossy surfaces [20,21,14], symmetric reflectance functions [1,19,25], reflectance mixtures [18], and uncalibrated environment map lighting [4]. Despite these generalizations, Lambertian photometric stereo remains useful because of its simplicity and allowance for uncalibrated acquisition, as well as being an analytical "stepping stone" for developing more comprehensive techniques.

In order to obtain accurate reconstructions with any photometric stereo technique, Lambertian or not, one must identify shadowed regions in the images. Most approaches for isolating shadows rely on using enough light sources such that every surface point is illuminated by at least two or three of them, and then detecting and discarding intensity measurements having low values. The number of images may be as few as three or four [10,3,16] but can also be many more [31,30]. Since these methods detect shadows by analyzing the intensities at individual pixels, they can be unreliable when a surface has texture with low albedo, and when cast shadows prevent some surface points from being illuminated by a sufficient number of lights.

An alternative approach is proposed by Chandraker et al. [8]. They estimate which light sources can be seen by each surface point using a Markov random field in which the per-pixel "data term" is based on Lambertian photometric stereo and the "smoothness term" acts to encourage spatial coherence. This approach requires that the light directions are calibrated and known, and like the methods above, relies on reasoning about the intensities at each pixel. Our approach also derives from Lambertian photometric stereo, but unlike [8], does not require the light sources to be calibrated. Moreover, instead of reasoning about per-pixel intensities, it reasons about illumination subspaces.

Our work is also related to the problem of characterizing the structure of the set of a scene's images. There exist bounds on the dimension of the image-span of convex Lambertian scenes under directional lighting [23] and environment map lighting [5,22], as well as convex scenes with a single arbitrary reflectance function [6] and mixtures of reflectance functions [13]. All of these bounds assume the scene to be convex so that cast shadows are absent. As a by-product of our analysis, we derive a complimentary bound that accommodates cast shadows and is valid for any Lambertian scene illuminated by a finite set of directional lights.

Finally, our work leverages insight from subspace clustering techniques, such as Generalized Principal Component Analysis (GPCA) [28] and Local Subspace Affinity (LSA) [32], that have been developed for motion segmentation. In our case, we perform subspace clustering using RANdom SAmple Consensus (RANSAC) [12,26,27]. This is quite different from a previous use of RANSAC in photometric stereo [17], which was aimed at identifying contour generators within an object's visual hull.

3 Visibility Subspaces

We begin with background and notation. For a Lambertian surface, the radiance from a surface point with normal $N \in \mathbb{S}^2$ and albedo ρ, illuminated with directional lighting L (i.e., with direction $L/||L|| \in \mathbb{S}^2$, and magnitude $||L||$), is given by $I = \max(0, \rho L^T N)$. In the absence of shadows, we know that $L^T N > 0$, and the image observations at m surface points illuminated by n light sources can be arranged as an $n \times m$ data matrix $\mathbf{I}$ that is the product of the $3 \times n$ lighting matrix $\mathbf{L} = [L_1, L_2, \cdots, L_n]$ and the $3 \times m$ albedo-scaled normals matrix $\mathbf{N} = [\rho_1 N_1, \rho_2 N_2, \cdots, \rho_m N_m]$:

$$\mathbf{I} = \mathbf{L}^T \mathbf{N}. \tag{1}$$

$\mathbf{L}$ and $\mathbf{N}$ are at most rank-three, and therefore, so is matrix $\mathbf{I}$ [29,23].

If the scene is imaged under at least three non-coplanar light sources and these sources are calibrated and known, the surface normals can be estimated from noisy image intensities as $\mathbf{N} = (\mathbf{L}^T)^+\mathbf{I}$, where $(\cdot)^+$ is the pseudo-inverse operator [29]. If the light sources are not calibrated, we can factor $\mathbf{I}$ using singular value decomposition (SVD) to recover the normals and lights using a rank-three approximation [15]:

$$\mathbf{I} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T, \quad \hat{\mathbf{L}}^T \triangleq \mathbf{U}_3\mathbf{\Lambda}_3^{\frac{1}{2}}, \quad \hat{\mathbf{N}} \triangleq \mathbf{\Lambda}_3^{\frac{1}{2}}\mathbf{V}_3^T. \tag{2}$$

This determines the normals up to a linear 3×3 linear ambiguity such that:

$$\mathbf{L}^T = \hat{\mathbf{L}}^T\mathbf{A}, \mathbf{N} = \mathbf{A}^{-1}\hat{\mathbf{N}}. \tag{3}$$

for some non-singular matrix $\mathbf{A}$. This ambiguity can be resolved if light source intensities or surface albedos are known [15]. It can also be resolved up to the three-parameter generalized bas-relief ambiguity by enforcing an integrability condition on the normal field [7,33].

Up to this point we have assumed the absence of cast and attached shadows, or equivalently, that every light source is *visible* to every surface normal. Now suppose that shadows exist, and consider the following toy example. A scene is partitioned into two uniform-visibility regions S_1 and S_2 that project to m_1 and m_2 pixels respectively. The scene is imaged under a set of n light directions that can be grouped into two (potentially) overlapping subsets $\mathbf{L}_1$ and $\mathbf{L}_2$, such that all of the lights $\mathbf{L}_1$ are visible to all points in S_1, and all of the lights $\mathbf{L}_2$ are visible to all points in S_2. Let the number of lights in these overlapping subsets be denoted by n_1 and n_2, and since they might overlap, we have $n_1 + n_2 \geq n$.

Now, the data matrix $\mathbf{I}$ can be permuted so that the first m_1 columns correspond to S_1 and last m_2 columns to S_2, and the first n_1 rows correspond to $\mathbf{L}_1$ and last n_2 rows to $\mathbf{L}_2$ with their shared lights lined up in the middle. Then, the observation matrix can be written as two sub-matrices, and if we denote by $\mathbf{N}_k$ the collection of surface normals in region S_k, the matrix can be factored as:

$$\mathbf{I} = [\ \mathbf{I}_1 \mid \mathbf{I}_2\ ] = \left[\begin{array}{c|c} \mathbf{L}_1^T & \mathbf{0}_{n-n_2}^T \\ \mathbf{0}_{n-n_1}^T & \mathbf{L}_2^T \end{array}\right] \left[\begin{array}{cc} \mathbf{N}_1 & \mathbf{0}_{m_2} \\ \mathbf{0}_{m_1} & \mathbf{N}_2 \end{array}\right], \tag{4}$$

with $\mathbf{0}_x$ representing a matrix of zeros with size $3 \times x$. The form of this factorization shows that while the row-space of $\mathbf{I}$ spans six dimensions, it actually consists of two rank-three subspaces corresponding to the two disjoint surface regions with different visibilities.

To generalize this to multiple regions with arbitrarily overlapping visibilities (i.e., sets of visible light sources), we define the *visibility vector* of region S_k to be the binary vector $V_k = [v_{k1}, v_{k2}, \cdots, v_{kn}]$, such that $v_{ki} = 1$ if light source L_i is visible to all the points in S_k and $v_{ki} = 0$ otherwise. The light sources visible to region S_k can then be expressed (with a slight change in notation from Eq. 4) as

$$\mathbf{L}_k = \mathbf{L} \otimes V_k, \tag{5}$$

where $\otimes$ represents the element-wise Hadamard product applied to every row of the lighting matrix. As above, we can then factor the observation matrix for a scene with s distinct visibility regions as:

$$\mathbf{I} = [\, \mathbf{I}_1 \mid \mathbf{I}_2 \mid \cdots \mid \mathbf{I}_s \,] = [\, \mathbf{L}_1^T \mid \mathbf{L}_2^T \mid \cdots \mid \mathbf{L}_s^T \,] \begin{bmatrix} \mathbf{N}_1 & & & \\ & \mathbf{N}_2 & & \\ & & \ddots & \\ & & & \mathbf{N}_s \end{bmatrix}, \qquad (6)$$

where $\mathbf{N}_k$ is the surface normal matrix corresponding to region S_k.

Thus, the observation matrix is made up of multiple subspaces, and we call these *visibility subspaces* because they correspond to regions in the scene that each have a consistent set of visible lights. Clearly, each subspace is at most rank-three, and the row space of a scene with s visibility subspaces has dimension at most $3s$. This leads us to the following:

Proposition. *The set of all images of a Lambertian scene illuminated by any combination of n directional light sources lies in a linear space with dimension at most $3 \cdot 2^n$.*

Proof: A scene illuminated by n light sources will have at most 2^n regions with distinct visibility configurations. The images of each region span at most a three-dimensional space, so the dimension of the image-span of the entire scene is at most $3 \cdot 2^n$.

This result is complementary to previous work that has established bounds on the dimensionality of scene appearance. Belhumeur and Kriegman [6] showed that the images of a scene with an arbitrary uniform BRDF, and illuminated by distant (environment map) lighting, lie in a linear space whose dimension is bounded by the number of distinct surface normals in the scene. Garg et al. [13] generalized this to spatially-varying reflectances that can be expressed as a linear combination of basis BRDFs. However, these results apply only to convex scenes without attached or cast shadows. In addition, these results assume that there are a finite number of normals in the scene to derive a bound on the dimensionality of scene appearance under arbitrary directional (environment map) lighting. In contrast, our analysis provides bounds on the appearance of a Lambertian scene with an arbitrary number of normals but illuminated by a finite number of light sources, and allows any form of shadowing.

In general, we do not know the visibility subspaces of a scene *a priori*, and we cannot permute the rows and columns of the observation matrix to directly obtain the factorization in Eq. 6. However, as we show next, we can identify the subspaces automatically using a subspace clustering technique.

4 Estimating Visibility Subspaces

RANSAC [12] is a statistical method for fitting models of known dimensions to data with noise and outliers. While RANSAC is traditionally used to discard

outliers from a dataset, we follow [27] and use it to cluster subspaces. In this context, it can be seen as an alternative to other subspace-estimation techniques, such as GPCA [28] and LSA [32].

Each visibility subspace of the scene is contained in a three-dimensional space. If we randomly choose three surface points that happen to be in the same region S_k, the light estimates $\hat{\mathbf{L}}_k$ that we obtain by factoring the image intensities at these three points (using Eq. 2) will accurately explain the intensities for all pixels in S_k. Thus, we expect a large number of "inliers". (Of course, there will be outliers as well because the points in the remainder of the scene will not have the same set of visible lights, and projecting their intensities onto $\hat{\mathbf{L}}_k$ will produce large errors.) Conversely, if we happen to choose three scene points that are in different regions, the light directions obtained by SVD will be unlikely to accurately explain the intensities at many other scene points, and we expect the number of inliers to be small. These observations suggest the following algorithm:

1. Choose three pixels at random and factor their intensities as $\mathbf{I}_3 = \hat{\mathbf{L}}_3^T \hat{\mathbf{N}}_3$.
2. Use lights $\hat{\mathbf{L}}_3$ to estimate the normal at all the surface points as $\hat{N}_i = (\hat{\mathbf{L}}_3^T)^+ I_i$.
3. Compute the per-pixel error of the estimated lights and normals as $E_i = ||I_i - \hat{\mathbf{L}}_3^T \hat{N}_i||^2$.
4. Mark points with error $E_i < \epsilon$ as inliers and recompute the associated optimal lighting $\hat{\mathbf{L}}_k$ using intensities for all inliers.
5. Repeat steps 1 through 4 for t iterations, or until a sufficiently large set of inliers has been found. During these iterations, keep track of the largest set of inliers found.
6. Mark the largest set of points that are inliers as a valid visibility subspace S_k with associated lighting basis $\hat{\mathbf{L}}_k$. Remove these inliers from the point set, and repeat steps 1 to 5 until all visibility subspaces have been recovered.

This procedure samples the points in the scene to find three points that belong to the same visibility subspace. Each time the sampling is successful, as measured by the number of inliers in Step 4, it extracts the subspace and removes it from the set of unlabeled points. The algorithm does not depend on the scene geometry or the lighting directions; it depends only on the rank-three condition of any visibility subspace. The result of the procedure is the set of per-pixel surface normals $\hat{\mathbf{N}}$, the per-pixel subspace labels $\mathbf{S}$, and a redundant (per-subspace) set of estimates for the light directions $\{\hat{\mathbf{L}}_k\}$. Note that in an uncalibrated setting, the set of normals for each subspace and their corresponding lights $\hat{\mathbf{L}}_k$ are defined up to their own linear ambiguity per Eqs. 2 and 3.

In our experiments, we use $t = 1000$ iterations, set the error threshold ϵ according to the noise in the input images, and run the procedure until 99% of the pixels are assigned to a valid visibility subspace. The remaining 1% of pixels are assigned to the subspace that best explains their intensity variation.

4.1 Degenerate Subspaces

The RANSAC-based method described above assumes that all visibility subspaces have rank-three. This is valid for any region having at least three

non-coplanar surface normals, and illuminated by at least three non-coplanar light sources. However, in general, scenes may contain rank-deficient subspaces that corrupt the clustering. Under the assumption that every point in the scene sees at least three non-coplanar lights (without which surface normal recovery is ambiguous), a visibility subspace can only be rank-deficient if it has degenerate normals: a region with coplanar normals will have rank two and a planar region will have rank one. Our task, then, is to check our recovered rank-three subspaces to see if they are composed of smaller degenerate subspaces.

Given the form of the observation matrix factorization in Eq. 6, it follows that a rank-three subspace can only be one of the following three types:

1. A region with a single visibility vector and non-coplanar normals (i.e., a true rank-three subspace).
2. Two regions with distinct visibility vectors, where one region has coplanar normals, and the other is planar (i.e., a combination of rank-two and rank-one subspaces).
3. Three regions with distinct visibilities, each of which is planar (i.e., a combination of three rank-one subspaces).

To ensure that our subspaces estimated by RANSAC are not of Type 2 or Type 3, we test every estimated rank-three subspace by searching for embedded rank-two and rank-one subspaces. If the number of pixels corresponding to the smaller embedded subspaces subsume more than a fraction α of the original set ($\alpha = 0.5$ in our experiments) we relabel them as being members of a different rank-deficient subspace.

5 Subspaces to Surface Normals

This subspace clustering identifies surface regions with uniform visibility, but does not provide a clean visibility vector V_k (or accurate shadows) for each region. Put another way, the non-visible entries of each $\hat{\mathbf{L}}_k$ are not necessarily zero-valued. To recover the visibility vectors and refine the light matrices, we separately examine the light estimates in each subspace $\hat{\mathbf{L}}_k = [\hat{L}_{k1}, \hat{L}_{k2}, \cdots, \hat{L}_{kn}]$, and provided that the subspace is not degenerate, we set

$$v_{ki} = ||\hat{L}_{ki}^T|| > \tau, \tag{7}$$

with $\tau = 0.25$ in our experiments. This simple approach succeeds because the normals $\hat{\mathbf{N}}_k$ in each non-degenerate subspace span three dimensions, so the product $I_{ki} \approx \hat{L}_{ki}^T \hat{\mathbf{N}}_k$ can be zero only if the light strength $||\hat{L}_{ki}||$ is zero. Effectively, we are able to recover the visibility for each subspace by reasoning about the magnitude of the subspace lighting—an approach that is independent of scene albedo and is, therefore, not confounded by texture.

To estimate the visibility for degenerate subspaces, we first project the subspace lighting onto the column-space of the subspace normals before thresholding their magnitudes. This removes the component of the lighting orthogonal to the

subspace normals that could be arbitrarily large while not contributing to the observed intensities.

Once the visibility vector for each subspace is known, we can recover the surface normals and reconstruct the surface. In the calibrated case, this is quite straightforward. Since the light sources $\mathbf{L}$ are known, they are combined with the visibility vectors using Eq. 5, and then the normals in every subspace are given by:

$$\mathbf{N}_k = (\mathbf{L} \otimes V_k)^+ \mathbf{I}_k \ , \ k = 1 \dots s. \tag{8}$$

If the light sources are *not* calibrated, the situation is more complex because the subspace clustering induces a distinct linear ambiguity in each subspace, (i.e., $\mathbf{L}_k^T = \hat{\mathbf{L}}_k^T \mathbf{A}_k, \mathbf{N}_k = \mathbf{A}_k^{-1} \hat{\mathbf{N}}_k, k = 1 \dots s$). Recovering the entire surface up to a single global ambiguity $\mathbf{A}$, which is the best we can do without additional information, requires that we somehow determine the transformations–one per subspace–that map each set of normals to a common coordinate system. Fortunately, this can be achieved by solving the set of linear equations:

$$\hat{\mathbf{L}} \otimes V_k = \hat{\mathbf{L}}_k \mathbf{A}_k^T \ , \ k = 1 \dots s, \tag{9}$$

where both the global lights $\hat{\mathbf{L}}$ (i.e., those defined up to a single global ambiguity) and the per-subspace ambiguity matrices $\mathbf{A}_k$ are unknown. This is an over-constrained homogeneous system of linear equations since, for n lights and s subspaces, it contains $3ns$ constraints and $3n + 9s$ unknown variables. To avoid the trivial solution $\hat{\mathbf{L}} = \mathbf{A}_k = 0$ we set the ambiguity matrix for one reference subspace (chosen to be the non-degenerate subspace with the largest number of visible lights) to be the identity matrix. Accordingly, we recover the global lights $\hat{\mathbf{L}}$ and normals $\hat{\mathbf{N}}$ up to a single 3×3 ambiguity, which is that of the reference subspace.

To handle degenerate subspaces in the uncalibrated case, we first solve Eq. 9 using all non-degenerate subspaces, and as long as all of the global lights are visible to at least one of these regions, we can recover all of them. We then use these "auto-calibrated" lights to solve for the normals in the degenerate rank-one and rank-two subspaces using Eq. 8.

As a final step in the uncalibrated scenario, we may reduce or eliminate the global ambiguity using additional constraints, such as integrability of the normal field [7,33], specular or glossy highlights [14,11,25], interreflections [9], or a prior model of object albedo [2,24]. Then, in either calibrated or uncalibrated conditions, the estimated normals can be integrated to recover scene depth. In this integration process, one may optionally enforce the depth constraints that are induced by the visibility vectors and lights, and an elegant procedure for doing so can be found in [8].

6 Results

We evaluate the uncalibrated instantiation of our approach on two synthetic datasets and two captured datasets. In each case, we automatically cluster subspaces, determine visibility vectors, and compute lights and surface normals up

to a global 3×3 linear ambiguity. As mentioned above, there are ways to resolve this ambiguity, and since this is not the focus of this work, we simply do so by manual intervention.

For synthetic examples, we evaluate the recovered normals, lights, and visibility subspaces by comparing them to the ground-truth values that are used to synthesize the input images. For the captured examples, the "true" values for comparison are obtained as follows. First, we acquire a dense set of calibrated photometric stereo images using approximately 50 different light directions. From such a dense set of calibrated images, we can robustly estimate surface albedos, and the image intensities can be reliably thresholded to detect per-pixel shadows and "true" visibilities. Then, we discard the shadowed measurements and recover the "true" normals via calibrated Lambertian photometric stereo. To make a direct comparison between this ground truth and our results, we execute our algorithm using a small subset of the dense input images, with the calibration information held out.

Figure 2 is a synthetic example in which the attached and cast shadows induce intricate visibility subspaces. From the six input images, our approach recovers the visibilities and normals almost perfectly. Figure 3 is a similar example, but in this case, the shadows cast on the back plane create degenerate visibility subspaces. These degenerate rank-one and rank-two subspaces are successfully detected by our approach, and the final visibilities and normals computed from the seven input images are again very close to ground truth. The median angular errors in surface normals for these two examples are $0.49°$ and $0.51°$, respectively. Note that both of these synthetic scenes have high-frequency texture and large variations in albedos. These conditions often lead to poor results when using intensity-based shadow detection from such a small number of images, but this is not the case for the proposed method.

In the two captured datasets we consider – the frog (Fig. 4) and scholar (Fig. 6) sequences – our algorithm was given 8 and 12 input images, respectively. For each

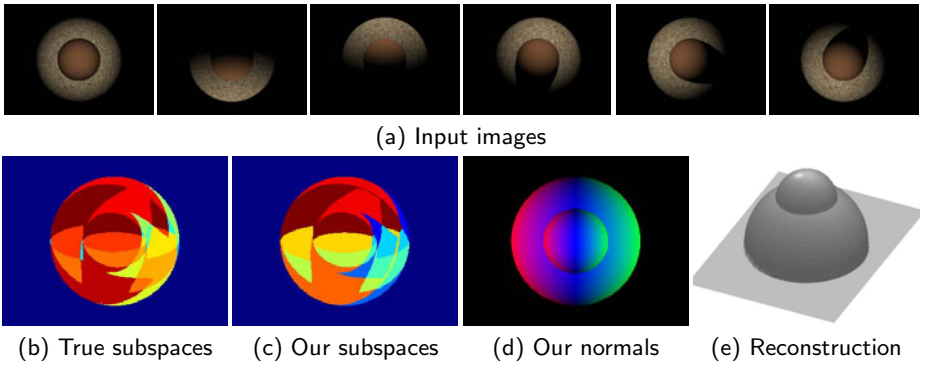


Fig. 2. Spheres sequence. Attached and cast shadows divide this scene into intricate visibility subspaces (b). We are able to recover them almost perfectly (c), and estimate the surface normals (d) and depth (e) accurately.

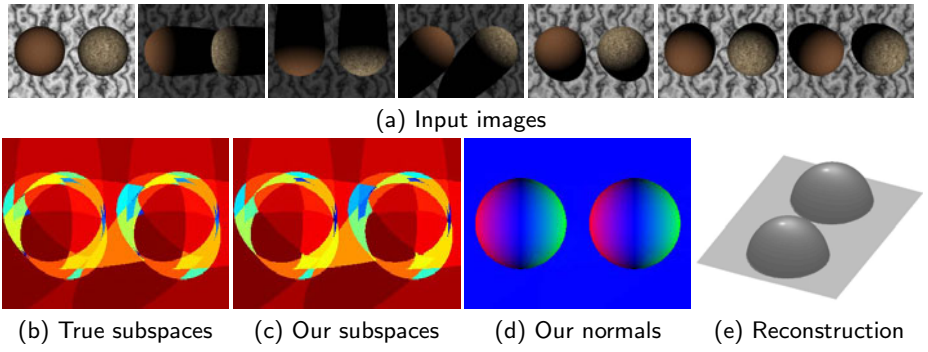


Fig. 3. Spheres and plane sequence. The shadows cast by the spheres on the plane create degenerate subspaces (b). We are able to disambiguate them and recover the visibility subspaces (c) and surface normals (d), and reconstruct the scene (e).

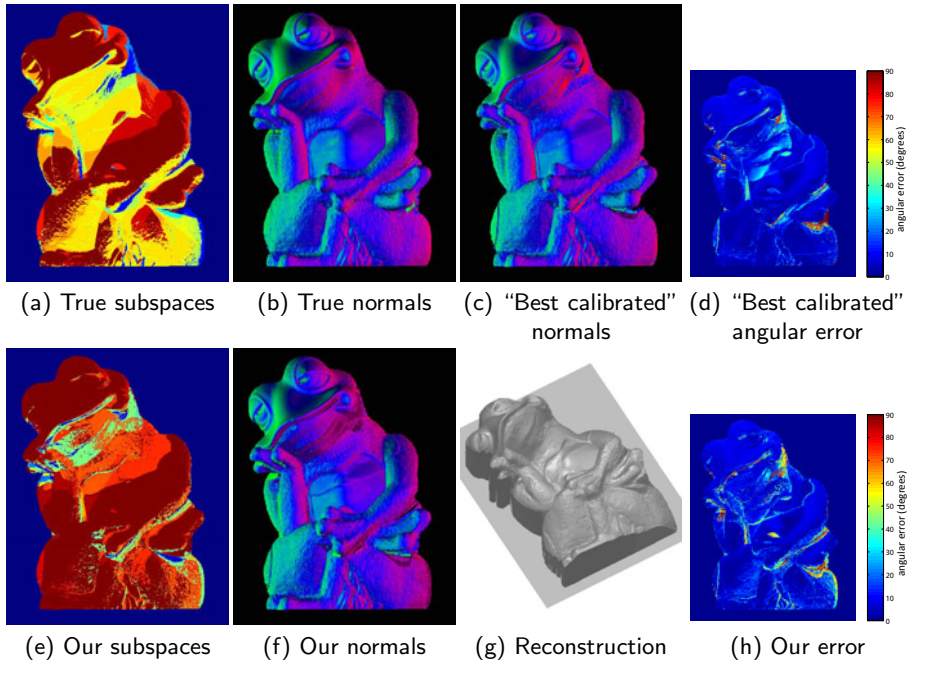


Fig. 4. Frog dataset. Reconstruction results from sparse input images (shown in Fig. 1). Despite slight specularity and convexities with mutual illumination, our estimated subspaces (e) match the ground truth (a) reasonably well. The angular differences between our normals (f) and ground truth normals (b) are most significant in regions having few non-shadowed measurements (h). For comparison, the normals estimated using calibrated photometric stereo equipped with perfect shadow detection (c) exhibit similar deviations from the ground truth (d).

of these datasets, we compare to the "true" normals and visibilities obtained from densely-sampled calibrated images as described above. We also compare the normals to those obtained using calibrated Lambertian photometric stereo applied to the same smaller set of (8 and 12) images that are available to our algorithm. We give this algorithm access to both the calibrated light directions as well as the ground truth visibilities. We refer to these normals as the "best calibrated" normals because they can be interpreted as calibrated Lambertian photometric stereo supplied with "perfect" shadow detection, or equivalently, as the best-possible result from a calibrated shadow-detection method, such as [8,30] applied on this small set of input images.

The input images have significant cast and attached shadows, and they exhibit non-idealities such as mutual illumination and slight specularity. Despite this, our method does reasonably well at locating the visibility subspaces (and shadows) from a small number of images. The median angular errors in the estimated normals (relative to the ground truth) are 7.44° and 4.45° for the frog and scholar datasets, respectively. The largest errors are made in regions with few non-shadowed measurements and where mutual illumination is most significant. This is not unique to our approach, however, and the errors from calibrated Lambertian photometric stereo with perfect shadow detection have a very similar structure. This suggests that our approach, which automatically handles shadows and is uncalibrated, introduces limited additional errors compared to an ideal calibrated algorithm.

7 Conclusion

We formulate shadow-detection in Lambertian photometric stereo as a subspace clustering task. This avoids heuristic reasoning about the intensities at individual pixels, and it allows handling cast and attached shadows in uncalibrated conditions when only a small number of input images are available. In addition, we derive a bound on the dimension of the image-span of a Lambertian scene under a discrete set of lights, and this bound has the rare property of incorporating arbitrary shadowing.

Unlike many previous approaches to shadow detection [8,16], ours does not impose a preference for spatial coherence while detecting shadow regions. Indeed, we find that subspace clustering naturally leads to relatively coherent regions without this imposition. It is quite likely, however, that incorporating a spatial coherence constraint during subspace clustering could improve the results, especially in the presence of non-idealities like mutual illumination, and this may be a fruitful direction for future research.

Also, we have restricted ourselves to Lambertian scenes illuminated by directional lights, and it is worth considering how this analysis can be extended to handle more general conditions. In particular, one might consider general environment map lighting [4], where a proper consideration of visibility would overcome the current (and severe) restriction to convex surfaces.

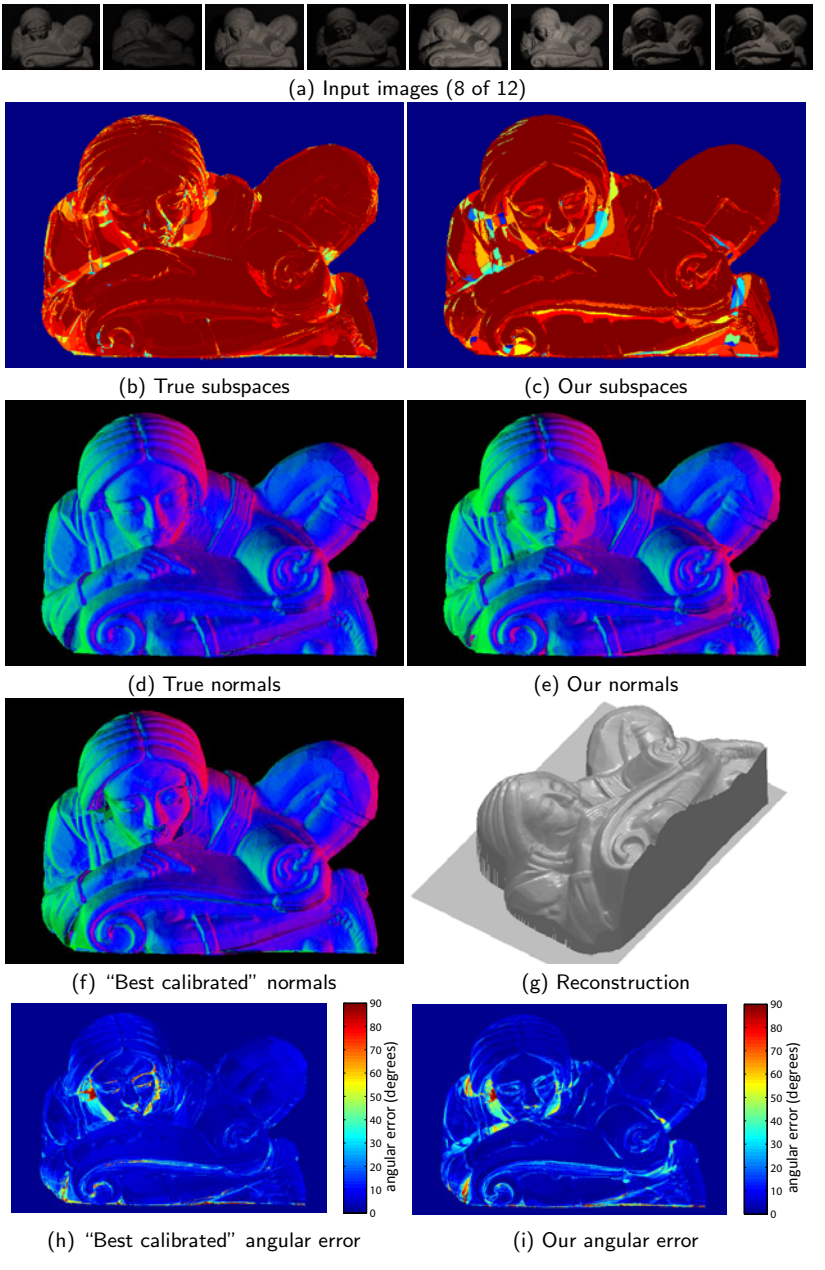


Fig. 5. Scholar dataset. The left column shows ground truth (b,d) and normals obtained by calibrated photometric stereo applied to sparse input images (f). Our results with the same sparse set of images (a) are shown in the right column (c,e,g). The angular differences between the true normals (d) and our estimates (e) show that most errors are small and that large errors are restricted to small regions with strong inter-reflections (i). For comparison, the calibrated result (f) also exhibits similar deviations (h).

Acknowledgments. Todd Zickler was supported by NSF Career Award IIS-0546408 and a fellowship from the Alfred P. Sloan Foundation.

References

1. Alldrin, N., Kriegman, D.: Toward reconstructing surfaces with arbitrary isotropic reflectance: A stratified photometric stereo approach. In: Proc. IEEE Int. Conf. Computer Vision (2007)
2. Alldrin, N., Mallick, S., Kriegman, D.: Resolving the generalized bas-relief ambiguity by entropy minimization. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2007)
3. Barsky, S., Petrou, M.: The 4-source photometric stereo technique for three-dimensional surfaces in the presence of highlights and shadows. IEEE Trans. Pattern Anal. Mach. Intell. 25, 1239–1252 (2003)
4. Basri, R., Jacobs, D., Kemelmacher, I.: Photometric stereo with general, unknown lighting. Int. Journal of Computer Vision 72(3), 239–257 (2007)
5. Basri, R., Jacobs, D.W.: Lambertian reflectance and linear subspaces. IEEE Trans. Pattern Anal. Mach. Intell. 25(2), 218–233 (2003)
6. Belhumeur, P.N., Kriegman, D.J.: What is the set of images of an object under all possible illumination conditions? Int. Journal of Computer Vision 28(3), 245–260 (1998)
7. Belhumeur, P.N., Kriegman, D.J., Yuille, A.L.: The bas-relief ambiguity. Int. Journal of Computer Vision 35(1), 33–44 (1999)
8. Chandraker, M., Agarwal, S., Kriegman, D.: Shadowcuts: Photometric stereo with shadows. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2007)
9. Chandraker, M., Kahl, F., Kriegman, D.: Reflections on the generalized bas-relief ambiguity. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2005)
10. Coleman, E., Jain, R.: Obtaining 3-dimensional shape of textured and specular surfaces using four-source photometry. Computer Graphics and Image Processing 18(4), 309–328 (1982)
11. Drbohlav, O., Chaniler, M.: Can two specular pixels calibrate photometric stereo? In: Proc. IEEE Int. Conf. Computer Vision (2005)
12. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. ACM Commun. 24(6), 381–395 (1981)
13. Garg, R., Du, H., Seitz, S.M., Snavely, N.: The dimensionality of scene appearance. In: Proc. IEEE Int. Conf. Computer Vision (2009)
14. Georghiades, A.: Incorporating the Torrance and Sparrow model of reflectance in uncalibrated photometric stereo. In: Proc. IEEE Int. Conf. Computer Vision (2003)
15. Hayakawa, H.: Photometric stereo under a light source with arbitrary motion. J. Opt. Soc. Am. 11(11) (1994)
16. Hernández, C., Vogiatzis, G., Cipolla, R.: Shadows in three-source photometric stereo. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 290–303. Springer, Heidelberg (2008)
17. Hernández Esteban, C., Vogiatzis, G., Cipolla, R.: Multiview photometric stereo. IEEE Trans. Pattern Anal. Mach. Intell. 30(3), 548–554 (2008)
18. Hertzmann, A., Seitz, S.M.: Example-based photometric stereo: Shape reconstruction with general, varying brdfs. IEEE Trans. Pattern Anal. Mach. Intell. 27, 1254–1264 (2005)

19. Holroyd, M., Lawrence, J., Humphreys, G., Zickler, T.: A photometric approach for estimating normals and tangents. ACM Trans. Graph. 27(5), 1–9 (2008)
20. Ikeuchi, K.: Determining surface orientations of specular surfaces by using the photometric stereo method. IEEE Trans. Pattern Anal. Mach. Intell. 3(6), 661–669 (1981)
21. Nayar, S., Ikeuchi, K., Kanade, T.: Determining shape and reflectance of hybrid surfaces by photometric sampling. IEEE T. Robotics Automation 6(4) (1990)
22. Ramamoorthi, R., Hanrahan, P.: On the relationship between radiance and irradiance: determining the illumination from images of a convex Lambertian object. J. Optical Society of America A 18(10), 2448–2458 (2001)
23. Shashua, A.: On photometric issues in 3D visual recognition from a single 2D image. Int. Journal of Computer Vision 31(1), 99–122 (1997)
24. Shi, B., Matsushita, Y., Wei, Y., Xu, C., Tan, P.: Self-calibrating photometric stereo. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2010)
25. Tan, P., Zickler, T.: A projective framework for radiometric image analysis. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2009)
26. Torr, P., Faugeras, O., Kanade, T., Hollinghurst, N., Lasenby, J., Sabin, M., Fitzgibbon, A.: Geometric motion segmentation and model selection (and discussion). Philosophical Transactions: Mathematical, Physical and Engineering Sciences 356(1740) (1998)
27. Tron, R., Vidal, R.: A benchmark for the comparison of 3-D motion segmentation algorithms. In: IEEE Conf. Computer Vision and Pattern Recognition (2007)
28. Vidal, R., Hartley, R.: Motion segmentation with missing data by powerfactorization and generalized PCA. In: IEEE Conf. Computer Vision and Pattern Recognition (2004)
29. Woodham, R.: Photometric stereo: A reflectance map technique for determining surface orientation from image intesity. In: Proc. SPIE, vol. 155, pp. 136–143 (1978)
30. Wu, T.P., Tang, C.K.: Photometric stereo via expectation maximization. IEEE Trans. Pattern Anal. Mach. Intell. 32(3), 546–560 (2010)
31. Wu, T.P., Tang, K.L., Tang, C.K., Wong, T.T.: Dense photometric stereo: A markov random field approach. IEEE Trans. Pattern Anal. Mach. Intell. 28(11), 1830–1846 (2006)
32. 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)
33. Yuille, A., Snow, D.: Shape and albedo from multiple images using integrability. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (1997)

Ring-Light Photometric Stereo

Zhenglong Zhou and Ping Tan

Department of Electrical & Computer Engineering, National University of Singapore

Abstract. We propose a novel algorithm for uncalibrated photometric stereo. While most of previous methods rely on various assumptions on scene properties, we exploit constraints in lighting configurations. We first derive an ambiguous reconstruction by requiring lights to lie on a view centered cone. This reconstruction is upgraded to Euclidean by constraints derived from lights of equal intensity and multiple view geometry. Compared to previous methods, our algorithm deals with more general data and achieves high accuracy. Another advantage of our method is that we can model weak perspective effects of lighting, while previous methods often assume orthographical illumination. We use both synthetic and real data to evaluate our algorithm. We further build a hardware prototype to demonstrate our approach.

1 Introduction

Photometric stereo algorithms [1] reconstruct local surface orientations (i.e. normal directions) from multiple images captured at a fixed viewpoint and variant illumination conditions. Most of these algorithms assume the illumination conditions are recorded during data capturing so that the normal directions are uniquely determined. However, capturing illumination conditions is often tedious, requiring the insertion of additional calibration objects such as mirrored spheres into the scene. These calibration objects can further cause inter-reflections that are often not modeled in photometric stereo algorithms and therefore increase reconstruction error.

There is a series of works [2,3,4,5,6,7,8,9,10,11] studying this problem without recording illumination conditions, known as uncalibrated photometric stereo. Almost all these methods rely on various assumptions about scene properties such as integrable surface, non-Lambertian materials, inter-reflections, six normals of equal albedo or small albedo entropy.[1] Hence, these methods can work for certain types of scenes that meet their assumptions, but cannot handle other types. For example, the gift box shown in Figure 1 (a) contains a few discrete planes with only three different normal directions and no significant non-Lambertian reflection. Notice that a plane does not provide integrable constraint as a linearly transformed plane is also integrable. Hence, all these previous methods will fail on this simple example. Figure 1 (b) is another challenging data which contains many depth discontinuities. Methods based on integrability must first identify

[1] An exception is Hayakawa's work [2] that used six lights with equal intensity to partially solve the problem.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 265–279, 2010.

these discontinuities which is a non-trivial task. Indeed previous uncalibrated photometric stereo algorithms mainly focused on a single segmented, smoothly curved object. Little work has been proposed to handle challenging data like those shown in Figure 1.

We propose to study uncalibrated photometric stereo by exploiting constrains in lighting configurations such that our method can be applied to more general data. We consider the case where a scene is illuminated by directional lights located on a view centered cone as illustrated in Figure 2 (a). We show that with at least five lights on such a cone, surface normal directions of a Lambertian scene can be recovered up to two kinds of rotations, and a scaling compounded with a mirror ambiguity. These ambiguities can be resolved if additional constraints are available, such as three lights of equal interval, five lights of equal intensity, surface integrability, non-Lambertian reflectance or corresponding normals from multiple viewpoints. To handle more general data, we choose to combine constraints derived from lighting configurations to achieve an Euclidean reconstruction. All we require about the scene is that two corresponding normals can be identified from two views, a constraint which can be easily satisfied for most inputs. We use synthetic and real data to evaluate our algorithm and build a prototype device to demonstrate potential applications.

2 Related Work

We first briefly review uncalibrated photometric stereo methods. Hayakawa [2] showed that surface normals can be recovered up to a general linear transformation if lighting directions are unknown. If one can identify six lights with equal intensity, or six normals with equal albedo, this general linear ambiguity can be reduced to a 3D rotation ambiguity. This approach can hardly handle surfaces with smooth varying texture or scenes with only a few different normals like the gift box example in Figure 1.

Most of the works in uncalibrated photometric stereo follow the seminal work by Belhumeur et al. [3] that proved the linear ambiguity can be reduced to a generalized bas-relief (GBR) ambiguity by surface integrability. Since then, many

(a) (b)

Fig. 1. Challenging data for uncalibrated photometric stereo. (a) is too simple and (b) is too complicate for most of existing methods.

works have been proposed to study and resolve this ambiguity. Drbohlav and Chantler [4,5] showed spike-specular reflectance can resolve the GBR ambiguity. Tan et al. [9,10] further proved any homogenous isotropic reflectance can resolve it. The GBR ambiguity can also be resolved by inter-reflections [6] and minimizing the entropy of surface albedos [8]. All these methods share a common limitation that depth discontinuities must be identified before integrability can be applied to obtain a reconstruction up to the GBR ambiguity. However, this identification of depth discontinuities is nontrivial in practice. Typically, a mask image is provided to separate the object from its background and the whole object surface is assumed to be integrable. This approach cannot handle complicated scenes like the one in Figure 1 (b). Furthermore, a piecewise planar scene, like Figure 1 (a), does not provide integrability constraints, because a plane is always integrable after any linear transformation. Hence, these algorithms often require a pre-segmented, smoothly curved surface.

Different from these previous works, we exploit partial information in the lighting conditions to resolve the shape ambiguity. Our method makes little assumption about the scene property. Hence, our method can be applied to more general data which cannot be handled by previous methods. Similar illumination configuration has been used [12] to minimize the reconstruction error due to camera sensor noise when lighting directions are known. Alldrin and Kriegman [13,14] also used the same configuration with known lighting. However, [13] recovers only partial surface geometry and [14] requires much more (about 100) input images. In comparison, our method requires only five images and our lighting directions are unknown. Our method is also related to those works that combine photometric stereo and structure-from-motion [15,16,17]. These methods assume the surface is differentiable and are difficult to be applied to complicated shapes like Figure 1 (b).

3 Ring-Light Photometric Stereo

Uncalibrated photometric stereo algorithms typically do not assume any prior knowledge about lighting conditions. In this section, we show that if the illumination is partially known, i.e. directional lights lying on a view centered cone, the problem can be significantly simplified. We first briefly review the shape ambiguity in uncalibrated photometric stereo. Then we show that lights lying on a view centered cone significantly reduce the ambiguity. At last, we describe several ways to resolve the remaining ambiguities.

3.1 Uncalibrated Photometric Stereo

We first briefly review the factorization based formulation of uncalibrated photometric stereo. Suppose F images are captured for a Lambertian surface under a variant directional lighting and each image contains P pixels. Ignoring shadows, inter-reflections and non-Lambertian effects, we can formulate the image intensity matrix I as $I = NL$. Here, I is a $P \times F$ matrix formed by pixel intensities. N and L are $P \times 3$ and $3 \times F$ matrices respectively. Each row of N indicates the

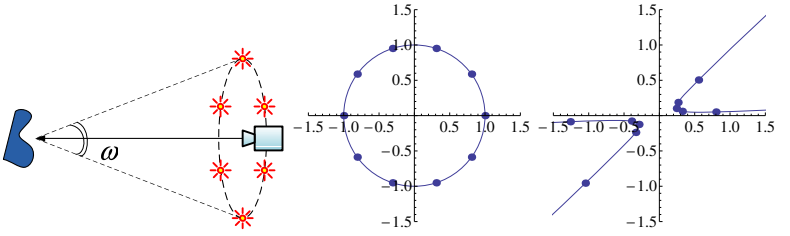


Fig. 2. Ring-light photometric stereo. (a) Lighting directions lie on a view centered cone. The term ω denotes the cone opening angle. (b) In the projective plane, these lights lie on a ring centered at origin (i.e. viewing direction). (c) When there is a linear ambiguity, these lights lie on a general planar conic. Our algorithm resolves this linear ambiguity by mapping lights back to their canonic positions.

scaled surface normal (unit surface normal multiplied with albedo), and each column of L is the scaled lighting direction (unit lighting direction multiplied with its intensity). In uncalibrated photometric stereo, only I is known and both N and L are unknown. Applying singular value decomposition (SVD), the matrix I can be decomposed as:

$$I = UDV^\top = (UD^{1/2})(D^{1/2}V^\top) = \hat{N}\hat{L}. \tag{1}$$

$\hat{N}, \hat{L}$ could differ from their true values by an arbitrary 3×3 invertible matrix A since $\hat{N}\hat{L} = \hat{N}A^{-1}A\hat{L}$. The autocalibration of photometric stereo amounts to recover A. Once A is estimated, the true surface normals and lighting directions can be computed as $N = \hat{N}A^{-1}$, $L = A\hat{L}$.

3.2 Constraints from a Ring-Light

Suppose the lights are distributed on a cone centered at the viewing direction as shown in Figure 2 (a). We follow the work [10] to analyze the problem in the projective plane where a lighting direction (l_x, l_y, l_z) is considered as a point $(l_x/l_z, l_y/l_z)$. We choose a world coordinate system such that the viewing direction is $(0, 0, 1)$ and corresponds to the origin in the projective plane. In the projective plane, the true lighting directions should lie on a circle centered at origin as shown in Figure 2 (b). This circle can be denoted by a diagonal matrix $C = diag(s^2, s^2, -1)$ and $C = S^\top C_u S$. Here, $C_u = diag(1, 1, -1)$ is the unit circle and $S = diag(s, s, 1)$ is a uniform scaling matrix. The SVD based reconstruction Equation (1) recovers lighting and normal directions up to an arbitrary invertible linear transformation A. The estimated lights form a general conic $\hat{C} = A^\top C A$ in the projective plane as shown in Figure 2 (c). Hence, we can resolve the ambiguity A by mapping $\hat{C}$ back to C. In this subsection, we first reduce the ambiguity by mapping $\hat{C}$ to the unit circle C_u. The remaining ambiguities are resolved in Section 3.3.

It is well known [18] that a conic can be computed from five points on it. Hence, we first use five estimated lighting directions to fit the conic $\hat{C}$ which

is a 3×3 symmetric matrix. We can apply SVD again to compute a linear transformation B that maps $\hat{C}$ to C_u, i.e.

$$\hat{C} = UDU^\top = (UD_1^{1/2})C_u(D_1^{1/2}U^\top) = B^\top C_u B.$$

Here, $D_1^{1/2}C_u D_1^{1/2} = D$. Then the lighting and surface normal directions can be updated accordingly by $\tilde{L} = B\hat{L}, \tilde{N} = \hat{N}B^{-1}$. Now, the general linear ambiguity is reduced and the estimated lights $\tilde{L}$ are on a view centered ring in the projective plane. But two kinds of ambiguities remain. First, the scaling matrix S between C and C_u is still unknown. Second, B can only be estimated up to a circle invariant transformation P that maps C_u to C_u. In other words, there could be an ambiguity matrix P such that $B^\top C_u B = B^\top P^\top C_u P B$. The following proposition specifies the structure of P.

Proposition 1: If a 3×3 linear transformation P maps the unit circle C_u to itself, i.e. $P^\top C_u P = C_u$, then P can be decomposed as $P = M^n R_\phi H_t R_\theta$, $n = 1 \text{ or } 2$. Here, M is a mirror transformation about the y axis, R_ϕ, R_θ are rotations in the plane (centered at origin), and H_t is a hyperbolic rotation, i.e.

$$M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, R_\theta = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}, H_t = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cosh t & \sinh t \\ 0 & \sinh t & \cosh t \end{pmatrix}. \quad (2)$$

R_ϕ has the same form as R_θ. Please refer to the appendix for a proof of this proposition. By this proposition, P is a compounded ambiguity that includes ordinary and hyperbolic rotations and a mirror transformation.

In the next section, we will discuss these ambiguities in more detail and propose methods to resolve them. Here we summarize these ambiguities by the following equation. The general conic $\hat{C}$ can be decomposed as:

$$\begin{aligned} \hat{C} &= B^\top C_u B = B^\top P^\top C_u P B \\ &= B^\top P^\top S^{-\top} C S^{-1} P B = A^\top C A \end{aligned} \quad (3)$$

Here, B is known, P and S are unknown transformations. Once P, S are determined, we can resolve the general linear ambiguity A. In the following, we refer to the compounded ambiguity $S^{-1}P$ as the *ring-light ambiguity*. It is also called the *ring-light transformation* depending on the context. The auto-calibration of ring-light photometric stereo amounts to estimate this compound transformation to upgrade the reconstruction $\tilde{L}, \tilde{N}$ to Euclidean as: $L = S^{-1}P\tilde{L}, N = \tilde{N}P^{-1}S$.

3.3 Ring-Light Ambiguities

We first briefly study each component of the *ring-light ambiguity* and later propose methods to solve it. Figure 3 summarizes these components and their geometric implications. The ambiguity S is a scaling in the projective plane which corresponds to the classic bas-relief ambiguity. M flips the normal and lighting directions vertically. It corresponds to the convex vs. concave ambiguity along the vertical direction. R_θ rotates the lighting and normal directions around the

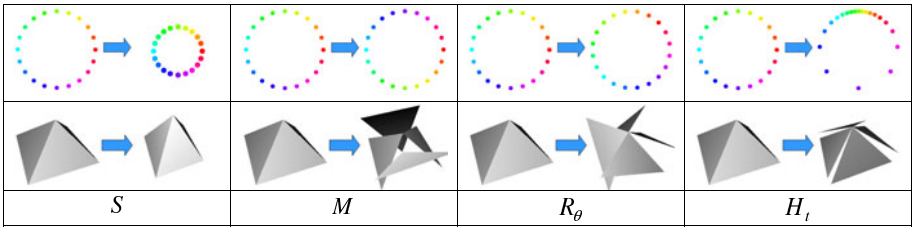


Fig. 3. Geometric explanations of the components of the 'ring-light ambiguity'. The first row shows the transformations induced to lighting directions in the projective plane. The second row illustrates the corresponding transformations to a 3D shape.

origin. It preserves all origin centered circles and could map a continuous shape to a discontinuous one. H_t is a hyperbolic rotation that preserves the unit circle. The relative positions of points on the unit circle are changed after a hyperbolic rotation as shown in Figure 3. It could also map continuous shapes to discontinuous ones.

In the following, we show various priors that resolve these ambiguities. We first discuss some widely used priors and later introduce three novel priors.

Integrability: Surface integrability is a widely used scene prior to resolve the ambiguity in uncalibrated photometric stereo. If the scene is known to be integrable, the linear ambiguity A can be reduced to a GBR ambiguity [3]. The intersection of the GBR transformation group with the ring-light transformation contains only the classic bas-relief transformation. Hence, if applicable, integrability resolves all the other components except the scaling S.

Points with Equal Albedo: Hayakawa [2] showed that six general normals with the same albedo can reduce the linear ambiguity to a 3D rotation compounded with a mirror reflection. The intersection of this ambiguity with the ring-light transformation contains only the planar rotation R_ϕ compounded with M. Hence, this prior reduces the ring-light ambiguity to a planar rotation with a mirror reflection.

Lights with Equal Intensity: Hayakawa's method [2] can also be applied to six general lights with equal intensity. However, since our lights lie on a view centered cone, constraints derived this way are degenerated. Both S, M and a 3D rotation cannot be resolved (explained in the next section). Hence, it reduces the ring-light ambiguity to a planar rotation R_ϕ compounded with a scaling S and a mirror M.

Lights with Equal Interval: If lights are uniformly distributed over the view centered cone, all lighting directions are determined up to a planar rotation (about the cone axis) and a scaling (corresponding to the unknown cone opening angle). Hence this constraint can reduce the linear ambiguity to a planar rotation R_ϕ compounded with a scaling S.

Multiple Viewpoint: Suppose a surface is observed from two different viewpoints with known relative motion and some corresponding points can be identified among these views. If the surface normals of both views are reconstructed up to some ambiguity, these corresponding points give constraints to resolve these ambiguities. In next section, we show that two corresponding normals from two views can resolve a planar rotation R_ϕ and a scaling S in both views.

Clockwise/Counter-Clockwise Lighting: M causes a vertical flipping of the estimated lighting and normal directions. If the lights on the ring are turned on one by one in clockwise or counter-clockwise, M reverses this order. Hence, M can be resolved if the order of lighting is known beforehand.

4 A Complete Stratified Reconstruction

We combine some of the discussed priors to achieve a Euclidean reconstruction. Those priors derived from lighting configurations are favored to handle more general scenes. We propose two methods to reduce the linear ambiguity to a planar rotation compounded with a scaling. In the next, we employ constraints derived from corresponding normals in different views to resolve the remaining ambiguities. For this stratified reconstruction, all we need are images from two viewpoints and five lights of equal interval/intensity distributed clockwise (or counterclockwise) on a view centered cone for each viewpoint.

4.1 Lights with Equal Interval

Suppose we know the order of lights (clockwise or counterclockwise). All lighting directions are determined up to the unknown cone opening angle and a planar rotation. We can assume arbitrary values of these two parameters to get pseudo lighting directions $\check{L}$ up to a scaling S (corresponding to the cone opening angle) and a planar rotation R_ϕ (corresponding to the rotation about the cone axis). We can recover normal directions up to the same ambiguity according to $\check{N} = I\check{L}^{-1}$. However, as we will see in experiments, this approach generates larger errors. Hence, we derive a more sophisticated approach in the following.

4.2 Lights with Equal Intensity

We first apply the ring-light constraint described in Section 3.2 to reconstruct normal directions up to a *ring-light ambiguity*. Then we apply the equal lighting intensity constraint to reduce the remaining ambiguities to a mirror transformation M, a planar rotation R_θ compounded with a scaling S. Afterwards, we use the known lighting order (clockwise in our experiments) to resolve M.

After applying the ring-light constraint, the estimated lighting direction $\tilde{\mathbf{l}}$ lies on the unit circle in the projective plane and is related to the true lighting direction $\mathbf{l}$ by $\mathbf{l} = S^{-1}P\tilde{\mathbf{l}}$. Suppose 5 lights are known to have equal intensity, we obtain

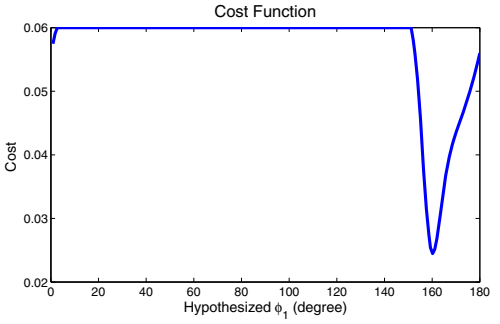


Fig. 4. The cost as a function of the hypothesized ϕ_1. This functions has a clear global minimum because ϕ_1, ϕ_2, s_1, s_2 are uniquely determined in principle.

$$\begin{aligned} k_1 &= \mathbf{l}_i{}^\top \mathbf{l}_i = \tilde{\mathbf{l}}_i{}^\top P^\top S^{-\top} S^{-1} P \tilde{\mathbf{l}}_i \\ &= \tilde{\mathbf{l}}_i{}^\top R_\theta{}^\top H_t{}^\top S^{-2} H_t R_\theta \tilde{\mathbf{l}}_i \qquad i = 1, 2, \cdots 5. \end{aligned} \tag{4}$$

It is easy to verify that M and R_ϕ are both eliminated from the equation. Here, k_1 is an unknown constant indicating the lighting intensity and i is an index of the lights. Let $F = R_\theta{}^\top H_t{}^\top S^{-2} H_t R_\theta$. Then Equation (4) is a linear equations about F, i.e. $\tilde{l}_i{}^\top F \tilde{l}_i = k_1$.

Hayakawa [2] used six such equations from different lighting directions to solve F. However, in our problem there are at most five independent linear equations because of the special configuration of lights. More specifically, $\tilde{\mathbf{l}}' \doteq H_t R_\theta \tilde{\mathbf{l}}$ must lie on the unit circle on the projective plane, because $\tilde{\mathbf{l}}$ lie on the unit circle which is invariant under H_t and R_θ. Hence, no matter what $S = diag(s, s, 1)$ is the expression, $\tilde{\mathbf{l}}_i{}^\top F \tilde{\mathbf{l}}_i = \tilde{\mathbf{l}}_i'{}^\top S^{-2} \tilde{\mathbf{l}}_i'$ is always a constant. In other words, S cannot be recovered from Equation (4) if these lights all lie on a view centered cone. To provide an experimental validation, we uniformly sample 360 lights on the unit circle. The six singular values of all these 360 equations are 17.72, 6.70, 6.70, 0.89, 0.63, 0.00. This suggests one degree of freedom of F cannot be determined.

Hence, we can only solve the 1D null space of F as $k_1 F_1 + k_2 F_2$. Here, F_1, F_2 satisfy $\tilde{\mathbf{l}}_i{}^\top F_1 \tilde{\mathbf{l}}_i = 1$ and $\tilde{\mathbf{l}}_i{}^\top F_2 \tilde{\mathbf{l}}_i = 0$ respectively, k_1 is the unknown but fixed constant and k_2 can vary to generate the whole 1D null space. We substitute $F = k_1 F_1 + k_2 F_2$ into $F = R_\theta{}^\top H_t{}^\top S^2 H_t R_\theta$. We solve s, t, θ, k_1 for any given k_2 according to the formulas provided in Appendix B. It can be verified that the solutions of t and θ are independent of k_2, while k_1 and s vary according to k_2. Hence, we obtain a unique solution of H_t and R_θ but cannot determine S, and the original *ring-light ambiguity* is reduced to M, S and R_ϕ. The result of this subsection is summarized into the following proposition.

Proposition 2: If five lights with equal intensity can be identified, the *ring-light ambiguity* can be reduced to a mirror transformation, a planar rotation compounded with a scaling.

4.3 Two Corresponding Normals in Two Views

We further exploit the constraints from multiple views. Suppose $\mathbf{n}_1$ and $\mathbf{n}_2$ are two corresponding normals in different views. They are defined in their local camera coordinate system and are related by the relative rotation between the two cameras, i.e. $\mathbf{n}_1 = T\mathbf{n}_2$. The relative rotation T can be computed separately, for example, by structure-from-motion. Suppose $\tilde{\mathbf{n}}_1, \tilde{\mathbf{n}}_2$ are the estimated normals which are subject to a planar rotation R_ϕ and scaling S. We have the following equations:

$$\mathbf{n}_1 \simeq S_1 R_{-\phi 1} \tilde{\mathbf{n}}_1 \qquad \mathbf{n}_2 \simeq S_2 R_{-\phi 2} \tilde{\mathbf{n}}_2 \qquad \mathbf{n}_1 = T\mathbf{n}_2. \tag{5}$$

Here, $\simeq$ means equal up to a scale. Hence,

$$\tilde{\mathbf{n}}_1 \simeq R_{\phi 1} S_1^{-1} T S_2 R_{-\phi 2} \tilde{\mathbf{n}}_2. \tag{6}$$

Let $E = R_{\phi 1} S_1^{-1} T S_2 R_{-\phi 2}$. We get $\tilde{\mathbf{n}}_1 \simeq E\tilde{\mathbf{n}}_2$. This equation provides two independent constraints. Hence, the four ambiguities $S_1, S_2, R_{\phi 1}, R_{\phi 2}$ can be resolved from two corresponding normals in two views.

Equation (6) can be written as $\tilde{\mathbf{n}}_1 \times E\tilde{\mathbf{n}}_2 = 0$, where $\times$ is the vector cross product. This vector equation expands to the following three equations:

$$s_2 \mathcal{A}^{(1)}(\phi_1, \phi_2) + \mathcal{B}^{(1)}(\phi_1) + s_1 s_2 \mathcal{C}^{(1)}(\phi_2) + s_1 \mathcal{D}^{(1)} = 0 \tag{7}$$

$$s_2 \mathcal{A}^{(2)}(\phi_1, \phi_2) + \mathcal{B}^{(2)}(\phi_1) + s_1 s_2 \mathcal{C}^{(2)}(\phi_2) + s_1 \mathcal{D}^{(2)} = 0 \tag{8}$$

$$s_2 \mathcal{A}^{(3)}(\phi_1, \phi_2) + \mathcal{B}^{(3)}(\phi_1) = 0. \tag{9}$$

Here, $\mathcal{D}^{(i)}$ are constants and $\mathcal{A}^{(i)}, \mathcal{B}^{(i)}$ and $\mathcal{C}^{(i)}$ are polynomials of trigonometrical functions of ϕ_1, ϕ_2.

$$\mathcal{A}^{(i)}(\phi_1, \phi_2) = a_1^{(i)} cos\phi_1 cos\phi_2 + a_2^{(i)} sin\phi_1 cos\phi_2 + a_3^{(i)} cos\phi_1 sin\phi_2 + a_4^{(i)} sin\phi_1 sin\phi_2$$

$$\mathcal{B}^{(i)}(\phi_1) = b_1^{(i)} cos\phi_1 + b_2^{(i)} sin\phi_1 \qquad \mathcal{C}^{(i)}(\phi_2) = c_1^{(i)} cos\phi_2 + c_2^{(i)} sin\phi_2$$

Here, $a_j^{(i)}, b_j^{(i)}$ and $c_j^{(i)}$ are all constants. These constants are provided in Appendix C.

Given two pairs of corresponding normals, it is nontrivial to derive an analytic solution for s_1, s_2, ϕ_1 and ϕ_2. We apply a 1D search for ϕ_1. For each hypothesized value of ϕ_1, ϕ_2 and s_2 can be easily solved from Equation (9) of both pairs. Then Equation (7) and Equation (8) from both pairs give totally 4 results for s_1. We use the consistency of these four values to choose the optimal ϕ_1 and its associated ϕ_2, s_2, s_1. In principle these four parameters are uniquely determined, so this 1D search has a global minimum and is robust as indicated in Figure 4. The result of this subsection is summarized in the following proposition.

Proposition 3: Given partial reconstructions of surface normals up to a planar rotation and a scaling from two views, if two pairs of corresponding normals can be identified, the reconstructions in both views can be upgraded to Euclidean.

5 Experiments

We apply our method to the challenging data shown in Figure 1. As explained earlier, these two examples cannot be handled by previous methods because they are either too simple (too few normals and planar surfaces) or too complicated (too many depth discontinuities). Our method first recovers a normal map up to the *ring-light ambiguity* as shown in the left column of Figure 5. Here, the x,y,z components of a normal direction are linearly encoded into the R,G,B color channels. This result is then upgraded to Euclidean by constraints derived from equal lighting intensity as shown in the middle. The right is a validation computed by calibrated photometric stereo where a metal sphere is used to record lighting directions. The difference between our results and the calibrated method is small. Some artifacts of the recovered normals on the box surface are due to the inaccuracy in radiometric calibration and inter-reflections.

Some additional results are shown in Figure 6. From left to right, we show one of the input images, our reconstructed surface normals and ground truth (obtained by calibrated photometric stereo). Some of the artifacts are due to non-Lambertian effects like shadow and highlight which are not modeled in our method. To handle shadows and highlights, we use simple intensity thresholding to exclude points with non-Lambertian effects. Our method is applied to Lambertian pixels to calibrate lighting directions. Then non-Lambertian pixels are processed with recovered lighting directions.

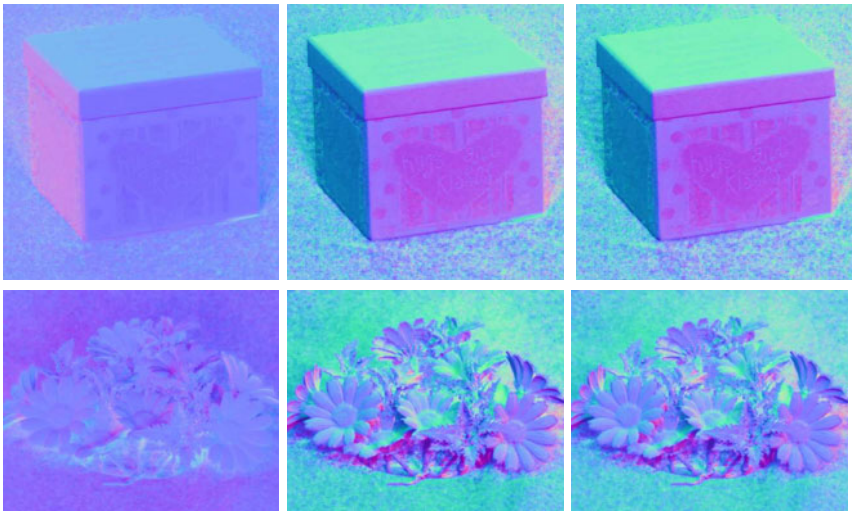


Fig. 5. Results for the challenging data in Figure 1. On the left are results up to the ring-light ambiguity. In the middle is our reconstructed surface normals. For a validation, we calibrate all incident lighting directions with a metal sphere and use calibrated photometric stereo to compute a ground truth in the right. Our result is very consistent to the ground truth.

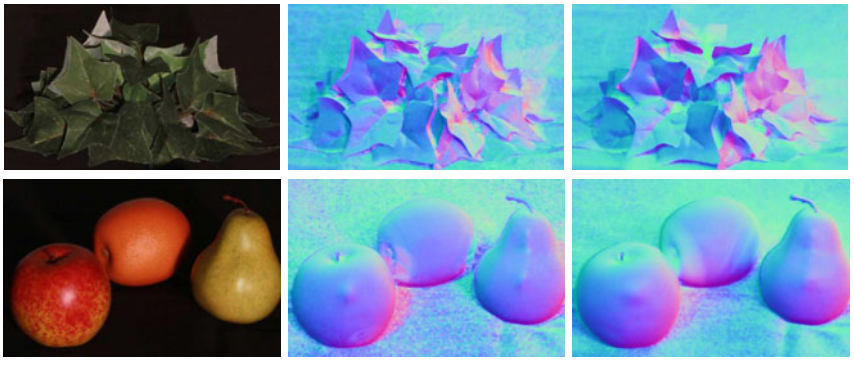


Fig. 6. Additional results. From left to right, they are one of the input images, our reconstructed surface normals, ground truth (by calibrated photometric stereo). Some of the artifacts are due to non-Lambertian effects like shadow and highlight.

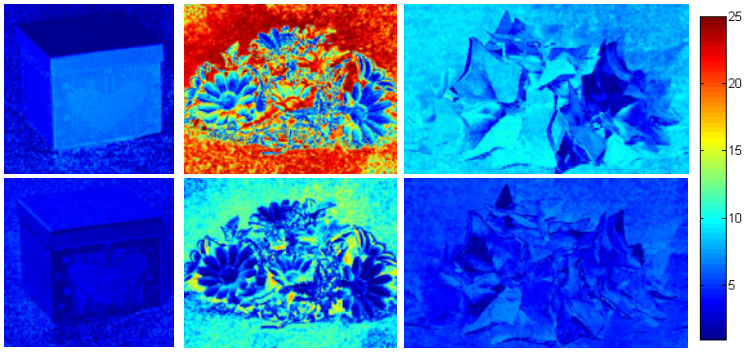


Fig. 7. The first and the second row are the angular errors of reconstructed normal directions by using equal lighting interval and equal lighting intensity constraint respectively. Typically, equal lighting intensity constraint generates more accurate results.

We also compare the two approaches to reduce the general linear ambiguity to a planar rotation and a scaling. The first and the second row of Figure 7 show the angular errors of reconstructed normal directions by equal lighting interval and equal lighting intensity constraints respectively. In the first row, the average angular errors are 5.8, 16.4 and 7.5 degrees from left to right. In the second row, these errors are 3.0, 6.0 and 4.4 degrees respectively. Please notice that normals in the background (a black cloth on table to reduce inter-reflection) are very noisy which increase the average angular error by 0.5-1 degrees in general. In our experiments, we find the constraints derived from lights with equal intensity are often more reliable. The flower example has larger error in both methods due to its strong shadowing and inter-reflection.

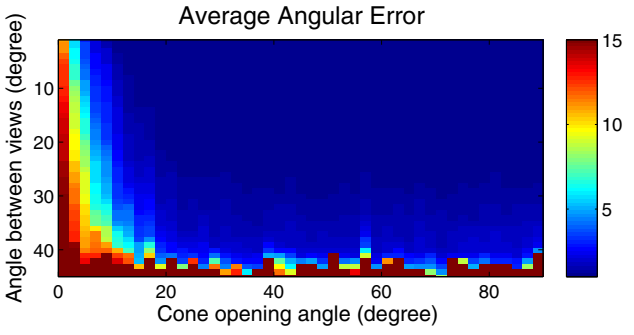
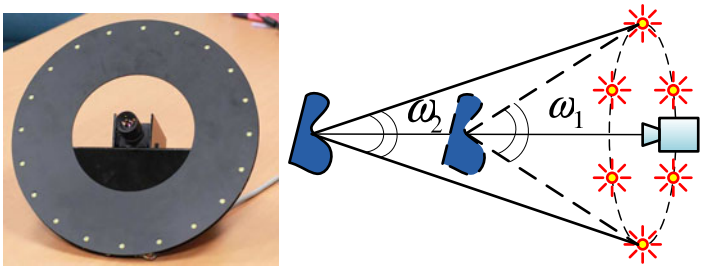


Fig. 8. Averaged angular error in the recovered normal directions as a function of the cone opening angle and the angle between two viewpoints. In most of time, the reconstruction error is smaller than 5 degrees.



(a) (b)

Fig. 9. Shown in (a) is a prototype device. 20 LED bulbs lie on a circle with radius of 150 millimeters centered at the viewing direction. Our method allows us to consider the weak perspective effects of the lighting which is critical for a handheld photometric stereo setup operating at relatively small distance. This weak perspective effects is illustrated in (b). To ensure the opening angle of the cone is larger than 10 degrees, the distance between the camera and captured objects should be within 1.7 meters.

Next, we use a synthetic scene containing two spheres to evaluate our system under various conditions. Images are synthesized at 840×560 resolution. The images are contaminated by Gaussian noise with zero mean and standard deviation 0.01 (pixel values are within [0,1]). We synthesize the scene from two viewpoints and at each viewpoint 10 lighting directions are generated on a view centered cone. Zero mean Gaussian noise with standard deviation of 0.5 pixel is added to the true corresponding pixel positions. We evaluate the reconstruction accuracy with respect to different values of the cone opening angle ω and the angle between the two viewing directions. The average angular error in reconstructed normal directions is shown in Figure 8. In most of the cases, the reconstruction is quite good with average error smaller than 5 degrees.

5.1 A Prototype Device

We manufacture a prototype device for ring-light photometric stereo according to our evaluation on synthetic data. The device is shown in Figure 9 (a) and consists 20 LED bulbs that are synchronized with a video camera to capture photometric stereo image sequences. The radius of the plate is 150 millimeters. Hence, according to Figure 8, the operation distance of the device should be less than 1.7 meters (cone opening angle larger than 10 degrees) to ensure the reconstruction accuracy. This device is similar to those handheld photometric stereo setups proposed in [15,16,17]. The advantage of our method is that our algorithm handles more general data and allows us to consider the weak perspective effects of lighting as illustrated in Figure 9 (b). We consider the lighting directions depend on the operation distance, e.g. $\omega_1 \neq \omega_2$. This effect is important when the operation distance is relatively small. A consequence is that this device cannot be pre-calibrated, because the incident lighting directions changes when the operation distance changes. For example, we pre-calibrate lighting directions for an operation distance of about 0.6 meters and apply it to the operation distance of about 0.8 meters. This incorrect pre-calibration causes average angular error on the box scene as large as 8.5 degrees. Almost three times larger than the 3.0 degrees error when our method is applied.

6 Conclusion

We have presented an stratified method for ring-light photometric stereo. We have shown that five lights on a view centered cone reduce the general linear ambiguity to two rotations, one mirror reflection compounded with a scaling. If these lights have equal intensity or equal interval, this compound *ring-light ambiguity* can be reduced to a planar rotation plus a scaling. If two corresponding normals from two viewpoints can be identified, Euclidean reconstruction can be obtained. Different from previous works on uncalibrated photometric stereo, we minimize the restriction on scene properties. Hence, our method can be applied to more general scenes. We also built a prototype device to demonstrate our method.

References

1. Woodham, R.: Photometric stereo: A reflectance map technique for determining surface orientation from image intensities. In: Proc. SPIE 22nd Annual Technical Symposium, pp. 136–143 (1978)
2. Hayakawa, H.: Photometric stereo under a light source with arbitrary motion. J. Optical Society of America A 11, 3079–3089 (1994)
3. Belhumeur, P.N., Kriegman, D.J., Yuille, A.L.: The bas-relief ambiguity. Int. Journal of Computer Vision 35, 33–44 (1999)
4. Drbohlav, O., Sara, R.: Specularities reduce ambiguity of uncalibrated photometric stereo. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2351, pp. 46–60. Springer, Heidelberg (2002)

5. Drbohlav, O., Chantler, M.: Can two specular pixels calibrate photometric stereo? In: Proc. ICCV, vol. 2, pp. 1850–1857 (2005)
6. Chandraker, M.K., Kahl, F., Kriegman, D.: Reflections on the generalized bas-relief ambiguity. In: Proc. of CVPR (2005)
7. Basri, R., Jacobs, D., Kemelmacher, I.: Photometric stereo with general, unknown lighting. Int. J. Comput. Vision 72, 239–257 (2007)
8. Alldrin, N., Mallick, S.P., Kriegman, D.J.: Resolving the generalized bas-relief ambiguity by entropy minimization. In: Proc. of CVPR (2007)
9. Tan, P., Mallick, S.P., Quan, L., Kriegman, D.J., Zickler, T.: Isotropy, reciprocity and the generalized bas-relief ambiguity. In: Proc. of CVPR (2007)
10. Tan, P., Zickler, T.: A projective framework for radiometric image analysis. In: Proc. of CVPR (2009)
11. Shi, B., Matsushita, Y., Wei, Y., Tan, P.: Self-calibrating photometric stereo. In: Proc. of CVPR (2010)
12. Drbohlav, O., Chantler, M.: On optimal light configurations in photometric stereo. In: Proc. ICCV (2005)
13. Alldrin, N., Kriegman, D.: Toward reconstructing surfaces with arbitrary isotropic reflectance: A stratified photometric stereo approach. In: Proc. ICCV (2007)
14. Alldrin, N., Zickler, T., Kriegman, D.: Photometric stereo with non-parametric and spatially-varying reflectance. In: Proc. of CVPR (2008)
15. Lim, J., Ho, J., Yang, M.H., Kriegman, D.: Passive photometric stereo from motion. In: Proc. ICCV (2005)
16. Joshi, N., Kriegman, D.: Shape from varying illumination and viewpoint. In: Proc. ICCV (2007)
17. Higo, T., Matsushita, Y., Joshi, N., Ikeuchi, K.: A hand-held photometric stereo camera for 3-d modeling. In: Proc. ICCV (2009)
18. Coxeter, H.S.M.: Introduction to Geometry, 2nd edn. Wiley, Chichester (1989)

A Appendix A: Proof of Proposition 1

Proposition 1: If a 3×3 linear transformation P maps the unit circle C_u to itself, i.e. $P^\top C_u P = C_u$, then P can be decomposed as $P = M^n R_\phi H_t R_\theta$,$n = 1$ or 2.

Proof: Our proof is based on the following two lemmas:

Lemma 1: If a conic C is mapped to another conic C' by a projective transformation P, then P maps the interior/exterior of C to the interior/exterior of C'.

Lemma 2: Suppose A and A' are two points on two different conics C and C'. B, B' lies inside of C, C' respectively. Then there are precisely two projective transformations which map C to C', A to A', and B to B'.

These lemmas can be found in [18]. In the following, for a general linear transformation P that maps C_u to C_u, we assume the pre-images of $(1, 0, 1)$ and $(0, 0, 1)$ are A and B respectively. We explicitly derive two transformations P_1, P_2, $P_1 \neq P_2$, with the form $M^n R_\phi H_t R_\theta$ that maps A, B to $(1, 0, 1)$ and $(0, 0, 1)$ respectively. Then according to Lemma 2, we know Proposition 1 is true.

According to the Lemma 1, B is a point within C_u. So we can denote B as $(rcos\theta, rsin\theta, 1)$, where $0 < r < 1$. It is easy to verify that $H_t R_{\pi/2-\theta}$ maps the

point B to the origin. Here, t is uniquely decided by $r = -sinh(t)/cosh(t)$. It is also easy to verify that $H_t R_{\pi/2-\theta}$ maps A to another point A' on the circle. We can denote A' as $(cos\phi, sin\phi, 1)$. Then a rotation $R_{-\phi}$ will maps A' to the point $(1, 0, 1)$ and keep the origin invariant. As a result, we get the following transformation $P_1 = R_{-\phi} H_t R_{\pi/2-\theta}$, that maps B to $(0, 0, 1)$ and A to $(1, 0, 1)$. Note that, we can define$P_2 = M R_{-\phi} H_t R_{\pi/2-\theta}$. P_2 should also maps B to origin and A to $(1, 0, 1)$. Further, $P_1 \neq P_2$. Hence, according to Lemma 2, they are the only two transformations that map A, B to $(1, 0, 1)$ and $(0, 0, 1)$ respectively.

B Appendix B: Determine t, s from F

θ can be directly computed from F, $\quad \theta = \arctan\left(-F_{13}/F_{23}\right)$
$k_1$can be solved from equation $(a^2 - b^2 - c^2)k_1^2 - (a + 3c)k_1 - 2 = 0$
where $a = \frac{1}{2}\left(F_{11} + F_{22}\right) + \frac{3}{2}F_{33} \quad b = \frac{1}{2}\left(F_{11} + F_{22} - F_{33}\right) \quad c = \frac{2F_{23}}{\cos\theta} = -\frac{2F_{13}}{\sin\theta}$
$s^{-2} = \frac{1}{2}\left(k_1(F_{11} + F_{22} - F_{33}) + 1\right)$
$t = \frac{1}{2} arcsinh\left(\frac{2k_1 F_{23}}{\cos\theta(s^{-2}+1)}\right) = \frac{1}{2} arccosh\left(\frac{k_1(F_{11}+F_{22}+F_{33})-s^{-2}}{s^{-2}+1}\right)$

C Appendix C: Constants in Equation 7–9

$$
\begin{aligned}
&T = \{t_{ij}\}_{3\times 3} \\
&a_1^{(1)} = -t_{21}n_{21}n_{13} - t_{22}n_{22}n_{13} \quad a_1^{(2)} = +t_{11}n_{21}n_{13} + t_{12}n_{22}n_{13} \\
&a_2^{(1)} = +t_{11}n_{21}n_{13} + t_{12}n_{22}n_{13} \quad a_2^{(2)} = +t_{21}n_{21}n_{13} + t_{22}n_{22}n_{13} \\
&a_3^{(1)} = -t_{22}n_{21}n_{13} + t_{21}n_{22}n_{13} \quad a_3^{(2)} = +t_{12}n_{21}n_{13} - t_{11}n_{22}n_{13} \\
&a_4^{(1)} = +t_{12}n_{21}n_{13} - t_{11}n_{22}n_{13} \quad a_4^{(2)} = +t_{22}n_{21}n_{13} - t_{21}n_{22}n_{13} \\
&a_1^{(3)} = +t_{21}n_{21}n_{11} + t_{22}n_{22}n_{11} - t_{11}n_{21}n_{12} - t_{12}n_{22}n_{12} \\
&a_2^{(3)} = -t_{11}n_{21}n_{11} - t_{12}n_{22}n_{11} - t_{21}n_{21}n_{12} - t_{22}n_{22}n_{12} \\
&a_3^{(3)} = +t_{22}n_{21}n_{11} - t_{21}n_{22}n_{11} - t_{12}n_{21}n_{12} + t_{11}n_{22}n_{12} \\
&a_4^{(3)} = -t_{12}n_{21}n_{11} + t_{11}n_{22}n_{11} - t_{22}n_{21}n_{12} + t_{21}n_{22}n_{12} \\
&b_1^{(1)} = -t_{23}n_{23}n_{13} \quad b_2^{(1)} = +t_{13}n_{23}n_{13} \quad b_1^{(2)} = +t_{13}n_{23}n_{13} \quad b_2^{(2)} = +t_{23}n_{23}n_{13} \\
&b_1^{(3)} = +t_{23}n_{23}n_{11} - t_{13}n_{23}n_{12} \quad b_2^{(3)} = -t_{13}n_{23}n_{11} - t_{23}n_{23}n_{12} \\
&c_1^{(1)} = +t_{31}n_{21}n_{12} + t_{32}n_{22}n_{12} \quad c_1^{(2)} = -t_{31}n_{21}n_{11} - t_{32}n_{22}n_{11} \\
&c_2^{(1)} = +t_{32}n_{21}n_{12} - t_{31}n_{22}n_{12} \quad c_2^{(2)} = -t_{32}n_{21}n_{11} + t_{31}n_{22}n_{11} \\
&c_1^{(3)} = c_2^{(3)} = \mathcal{D}^{(3)} = 0 \quad \mathcal{D}^{(1)} = +t_{33}n_{23}n_{12} \quad \mathcal{D}^{(2)} = -t_{33}n_{23}n_{11}
\end{aligned}
\tag{10}
$$

Shape from Second-Bounce of Light Transport

Siying Liu[1], Tian-Tsong Ng[1], and Yasuyuki Matsushita[2]

[1] Institute for Infocomm Research Singapore
[2] Microsoft Research Asia

Abstract. This paper describes a method to recover scene geometry from the second-bounce of light transport. We show that form factors (up to a scaling ambiguity) can be derived from the second-bounce component of light transport in a Lambertian case. The form factors carry information of the geometric relationship between every pair of scene points, *i.e.*, distance between scene points and relative surface orientations. Modelling the scene as polygonal, we develop a method to recover the scene geometry up to a scaling ambiguity from the form factors by optimization. Unlike other shape-from-intensity methods, our method simultaneously estimates depth and surface normal; therefore, our method can handle discontinuous surfaces as it can avoid surface normal integration. Various simulation and real-world experiments demonstrate the correctness of the proposed theory of shape recovery from light transport.

1 Introduction

Interreflections, reciprocal reflections among reflecting surfaces, are observed in all real-world scenes. The way light transports varies with scene geometry and surface reflectance. Clearly, there is a mutual dependency between the light transport and scene environment. This fact is used for scene modeling, *e.g.*, by Nayar [1] for scene geometry and reflectance, and also by Yu *et al.* [2] and Machida *et al.* [3] for modeling bidirectional reflectance distribution functions (BRDFs), when prior knowledge of the scene is available (pseudo geometry and reflectance for [1], and accurate scene geometry for [2,3]). Recent advances in computational photography enabled modeling of inverse light transport [4,5,6,7] from photographs without prior knowledge of the environment. These works open up a new open problem — can we infer scene geometry only from the light transport without any prior knowledge?

In this paper, we propose a new approach to inferring scene geometry from the measured light transport without using any prior knowledge about the scene. We focus our discussion on a Lambertian case and model the scene as composed of planar patches. Our approach can be viewed as an *inverse* radiosity method where the scene geometry is unknown a priori as illustrated in Fig. 1. We first show a form factor matrix, which represents how much light is transported from one scene point to another purely by geometric factor, up to a scaling ambiguity, can be obtained from the *second-bounce* of light transport. Using the form factor matrix, we show that scene geometry can be recovered up to a scaling ambiguity.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 280–293, 2010.

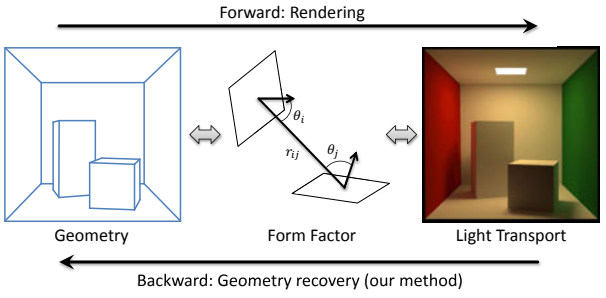


Fig. 1. The relationship between scene geometry and light transport. The forward case is a rendering process where light transport can be computed from known scene geometry, while the backward case is where the scene geometry is inferred from light transport.

We develop a solution method for simultaneously estimating surface orientations and scene depths from the form factor matrix by optimization.

The primary contributions of this paper are twofold. First, it introduces the use of the second-bounce of light transport for recovering scene geometry. We describe the relationship between the scene geometry and the light transport and show what information is carried in the second-bounce component about the scene. To this end, we show that the scene geometry can be recovered up to a scaling ambiguity as well as diffuse albedo ratios. Second, the proposed method is effective even when the surface of interest has discontinuity. Unlike prior shape-from-intensity methods, our method simultaneously estimates surface orientation and depth (up to scaling ambiguity). This allows us to avoid integration of surface orientations; therefore, the assumption of continuous surface is no longer needed unlike other shape-from-intensity methods.

1.1 Prior Work

Forward light transport is well studied in computer graphics such as in ray tracing [8] and radiosity [9,8]. These use known scene geometry and BRDFs for producing photorealistic images. More recently, photographic modeling of forward light transport is drawing attention [10,11,12,13]. These methods take a number of images under different lightings for recording various complex lighting effects.

In graphics, inverse global illumination was introduced by Yu *et al.* [14] for estimating reflectance properties, rather than for geometry estimation. Inverse light transport is also used in computer vision. Seitz *et al.* [4] showed a method for estimating n-bounce component of light transport by probing a scene using a narrow beam light. Ng *et al.* [7] extended the method using a stratified matrix inversion for radiometric compensation of projector-camera systems. Nayar *et al.* [5] proposed a fast method for separating direct and global component of light transport using high frequency illumination. Gupta *et al.* [6] later discussed the relation between global illumination and defocused illumination.

The goal of this paper is shape recovery from the measured light transport. The closest to our work is Nayar *et al.* [1]. They proposed an iterative photometric stereo algorithm, which is the first work that uses interreflections as a useful cue for shape and reflectance estimation. Our method is different from their approach in that we infer scene geometry directly from the second-bounce component of light transport instead of relying on photometric stereo. In addition, our method is not limited to continuous surfaces because our method does not require integration of the surface orientations, but simultaneously estimates surface orientation and depth. On the other hand, compared with their method, our method requires more images as input for obtaining the second-bounce component of the light transport.

Apart from shape-from-intensity methods, other prior art on shape or depth recovery include shape from structured light [15] and structure-from-motion (SfM) [16]. Both approaches use triangulation for determining depths. In terms of calibration requirements, these methods require calibration of intrinsic parameters of the imaging devices, while our method does not require intrinsic calibration.

Our method uses form factors for shape estimation. The computation of form factors has a long history back to Lambert in 1760 [17]. Schröder and Hanrahan derived a closed-form solution for the case of general polygons [18]. Our method uses form factors in an inverse manner for estimating the scene geometry.

2 Interreflection and Scene Geometry

2.1 Forward Case: The Rendering Equation

The rendering equation [19] is written as

$$L_{out}(\mathbf{p},\omega_o) = L_e(\mathbf{p},\omega_o) + \int_{M^2} \rho(\mathbf{p},\omega_i,\omega_o) L_{out}(\mathbf{p}',-\omega_i) V(\mathbf{p},\mathbf{p}') \frac{\cos\theta_i \cos\theta_o}{\|\mathbf{p}-\mathbf{p}'\|^2} dA_{\mathbf{p}'}, \tag{1}$$

where $L_{out}(\mathbf{p},\omega_o)$ is the reflected or outgoing radiance in direction ω_o, L_e is the emission corresponding to light sources, ρ is the Bidirectional Reflectance Distribution Function (BRDF) of the scene, and V is the binary visibility function. The visibility function $V(\mathbf{p},\mathbf{p}')$ is 1 if scene points $\mathbf{p}$ and $\mathbf{p}'$ are connected by a line of sight and 0 otherwise. The integral is over the area of M^2 of all scene surfaces, and weighted by a purely geometric factor known as the *form factor*.

The above rendering equation applies for a continuous surface. Discretization of the surface leads to a matrix representation. For a surface with n facets,[1] radiance and albedo values are assumed to be constant over each facet, then the rendering equation can be written in operator notation as [20]:

$$\mathbf{l}_{out} = \mathbf{l}_e + \mathbf{KG}\mathbf{l}_{out} = \mathbf{l}_e + \mathbf{A}\mathbf{l}_{out}, \text{ where } \mathbf{A} = \mathbf{KG}. \tag{2}$$

[1] In this paper, we use the term "facet" to describe the smallest piece of a surface subdivision and the term "patch" for any larger pieces, up to and including the biggest polygons formed by combining facets.

$\mathbf{l}_{out}$ is a vector of $L_{out}(\mathbf{p}, \omega_o)$, $\mathbf{l}_e$ is a vector of $L_e(\mathbf{p}, \omega_o)$, $\mathbf{G}$ is a purely geometric operator that takes outgoing or reflected radiance and propagates it within the scene to obtain incident radiance, and $\mathbf{K}$ is a local linear reflection operator based on the BRDF of the surface:

$$\mathbf{K} = \begin{bmatrix} \rho_1 & 0 & \cdots & 0 \\ 0 & \rho_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \rho_n \end{bmatrix}, \mathbf{G} = \begin{bmatrix} 0 & G_{12} & \cdots & G_{1n} \\ G_{21} & 0 & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ G_{n1} & \cdots & \cdots & 0 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 0 & \rho_1 G_{12} & \cdots & \rho_1 G_{1n} \\ \rho_2 G_{21} & 0 & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ \rho_n G_{n1} & \cdots & \cdots & 0 \end{bmatrix}. \tag{3}$$

The interreflections between two points or facets $\mathbf{p}_i$ and $\mathbf{p}_j$ can be described by the G_{ij} expression[2]:

$$G_{ij} = \frac{V(\mathbf{p}_i, \mathbf{p}_j) \cos\alpha \cos\beta}{\|\mathbf{r}_{ij}\|^2} = \frac{V(\mathbf{p}_i, \mathbf{p}_j)(-\widehat{\mathbf{r}}_{ij} \cdot \widehat{\mathbf{n}}_i)(\widehat{\mathbf{r}}_{ij} \cdot \widehat{\mathbf{n}}_j)}{\|\mathbf{r}_{ij}\|^2}, \tag{4}$$

where $\mathbf{r}_{ij} = \mathbf{p}_j - \mathbf{p}_i$, α and β are the angles between $\mathbf{r}_{ij}$ and their respective surface normals. G_{ii} is undefined for any i, and G_{ij} vanishes if $\mathbf{p}_i$ and $\mathbf{p}_j$ are mutually invisible.

2.2 Backward Case: From Light Transport T to form Factor G

Following [19] and Eq. (2), we can obtain

$$\mathbf{l}_{out} = (\mathbf{I} - \mathbf{A})^{-1} \mathbf{l}_e. \tag{5}$$

Assuming the camera does not see the light source directly, and we do not have emissive surfaces, we can replace $\mathbf{l}_e$ with the effective emission that corresponds to the direct reflection from the light source, $\mathbf{l}_e = \mathbf{F}\mathbf{l}_{in}$, where $\mathbf{l}_{in}$ is the incident light from a light source such as a projector, and $\mathbf{F}$ is the light transport matrix that corresponds to the first-bounce reflection. Assuming a focused light source, the first-bounce matrix $\mathbf{F}$ is diagonal. Hence, we have

$$\mathbf{l}_{out} = (\mathbf{I} - \mathbf{A})^{-1} \mathbf{F} \mathbf{l}_{in} = \mathbf{T} \mathbf{l}_{in}, \ \mathbf{T} = (\mathbf{I} - \mathbf{A})^{-1} \mathbf{F}, \tag{6}$$

where $\mathbf{T}$ is the light transport matrix. Hence, we can write $\mathbf{A}$ in terms of $\mathbf{T}$ as

$$\mathbf{A} = \mathbf{I} - \mathbf{F}\mathbf{T}^{-1}. \tag{7}$$

With Neumann series expansion, we can expand a light transport matrix into a matrix series where the second term corresponds to the second-bounce:

$$\mathbf{T} = (\mathbf{I} - \mathbf{A})^{-1} \mathbf{F} = (\mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \cdots) \mathbf{F}. \tag{8}$$

We can see that the matrix $\mathbf{A}$ is related to the second-bounce of light transport.

[2] In [1], the geometric kernel takes into account the effective area of the illuminator facet. Here, we assume facets i and j are interchangeably the illuminator and reflector and have sufficiently small area *s.t.* G_{ij} and G_{ji} are approximately equal.

According to [4], for a Lambertian scene, the diagonal elements of $\mathbf{F}$ are given by the reciprocals of the diagonal elements of $\mathbf{T}$:

$$\mathbf{F}[i,i] = \frac{1}{\mathbf{T}^{-1}[i,i]}. \tag{9}$$

Therefore, if we limit our discussion to the Lambertian case, $\mathbf{A}$ can be computed using Eq. (9) and Eq. (7). For geometry estimation, we can extract $\mathbf{G}$ from $\mathbf{A}$. Given $\mathbf{A}$, we can compute the relative albedo ρ_{ij} for all the scene points[3]:

$$\rho_{ij} \doteq \frac{A_{ij}}{A_{ji}} = \frac{\rho_i G_{ij}}{\rho_j G_{ji}} = \frac{\rho_i}{\rho_j}. \tag{10}$$

Given ρ_{ij}, we can recover $\mathbf{G}$ up to a scale:

$$\mathbf{A} = \tilde{\mathbf{K}}\tilde{\mathbf{G}} \text{ and } \tilde{\mathbf{G}} = \tilde{\mathbf{K}}^{-1}\mathbf{A}, \tag{11}$$

where

$$\tilde{\mathbf{K}} = \frac{1}{\rho_j}\mathbf{K} = \begin{bmatrix} \rho_{1j} & 0 & \cdots & 0 \\ 0 & \rho_{2j} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \rho_{nj} \end{bmatrix}, \quad \text{and} \quad \tilde{\mathbf{G}} = \rho_j \mathbf{G}. \tag{12}$$

3 Geometry Extraction from the Geometric Form Factors

For two mutually visible points[4] $\mathbf{p}_1$ and $\mathbf{p}_3$ as shown in Fig. 2 (a), the geometric form factor is given by $G_{13}(\mathbf{r}_{13}, \hat{\mathbf{n}}_1, \hat{\mathbf{n}}_3) = \frac{(-\hat{\mathbf{r}}_{13}\cdot\hat{\mathbf{n}}_1)(\hat{\mathbf{r}}_{13}\cdot\hat{\mathbf{n}}_3)}{\|\mathbf{r}_{13}\|^2}$. For mutually visibility, we need to have $(\hat{\mathbf{r}}_{13}\cdot\hat{\mathbf{n}}_1) > 0$ and $(\hat{\mathbf{r}}_{13}\cdot\hat{\mathbf{n}}_3) < 0$. Resolving $\hat{\mathbf{n}}_i$ and $\mathbf{r}_{ij}$ for all scene points recovers both depth and surface normal. However, this method is unable to recover geometry for scene points which are not visible by others, *e.g.*, geometry extraction is impossible for a globally convex surface. In this section, we will examine the settings under which the scene geometry can be extracted.

3.1 Problem Setup and Assumptions

In this work, we consider a light transport acquisition setup with a projector-camera system. Assuming no serious scattering due to the transmission media or subsurface scattering, the directional nature of the projector light allows correspondence between the projector pixels and scene points to be established. The directional light from the projector is perspective in nature. However, if we assume that the scene depth is small, the projection is approximately orthographic.

[3] As an observation, given the relative albedo ρ_{ij}, we can recover the absolute albedo value for all scene points as long as the absolute albedo value of one of the scene points is known.

[4] A discrete surface is composed of small facets that are often assumed to have uniform property. Hence, a discrete facet is conceptually similar to a discrete point. For ease of discussion, we may use the term "facet" and "point" interchangeably.

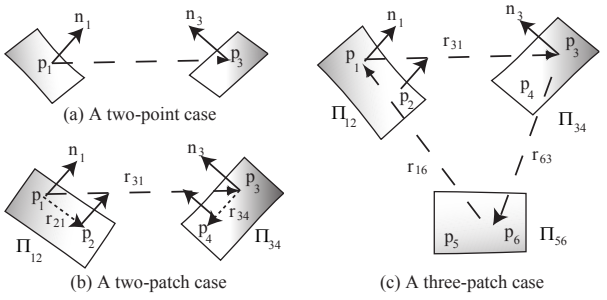


Fig. 2. Three different setup for geometry extraction from the geometric form factor terms

With this assumption, the problem of geometry extraction is greatly simplified, as we can assume that the correspondence points in the scene approximately preserve the grid structure of the projector pixels. In a coordinate frame where the z-axis is aligned with the optical axis of the projector, we can assume that the x-y coordinate of the scene points form a rectangular grid, which is known up to a scale, while the z-coordinate is the only unknown.

3.2 A Case with Two Scene Points

In the case with just two scene points as in Fig. 2 (a), knowing the value of the form factor is not sufficient to recover the surface normal nor the depth uniquely. To see the set of all possible solutions, we can rewrite Eq. (4) as

$$\widehat{\mathbf{r}}_{13} \cdot \widehat{\mathbf{n}}_1 = -\frac{G_{13}\|\mathbf{r}_{13}\|^2}{\widehat{\mathbf{r}}_{13} \cdot \widehat{\mathbf{n}}_3}. \tag{13}$$

In our setting, for the scene point position in $\mathbb{R}^3$, only the z-coordinate is unknown. As Eq. (13) can only be unique up to a relative depth in z-direction, there is no loss of generality to fix the z-coordinate for one of the scene point, say $\mathbf{p}_3$. Then, the distance vector is governed by z_1, *i.e.*, the z-coordinate of $\mathbf{p}_1$, alone. For every $\widehat{\mathbf{n}}_1$ in Eq. (13), it is possible to find a $\mathbf{r}_{13}$ for all $\widehat{\mathbf{n}}_3$. As $\widehat{\mathbf{n}}_1$ and $\widehat{\mathbf{n}}_3$ are unit normal vectors, they live in a spherical space $\mathbb{S}^2$. Hence, the space of all solutions $(\widehat{\mathbf{n}}_1, \widehat{\mathbf{n}}_3) \in \mathbb{S}^2 \times \mathbb{S}^2$, which is highly ambiguous. This also indicates that having more independent pairs of points does not help, as the normals are totally unconstrained.

3.3 A Case with Two Patches

To resolve the ambiguity in the form factor expression, we need to introduce more constraints. One way to do so is to group a set of adjacent points to form a patch where the points share a common surface normal. Fig. 2 (b) shows an example of two patches, each having two points. The newly introduced constraints are

$$\widehat{\mathbf{n}}_1 = \widehat{\mathbf{n}}_2, \quad \widehat{\mathbf{n}}_3 = \widehat{\mathbf{n}}_4, \quad \widehat{\mathbf{r}}_{12} \cdot \widehat{\mathbf{n}}_1 = 0, \text{ and } \widehat{\mathbf{r}}_{34} \cdot \widehat{\mathbf{n}}_3 = 0. \tag{14}$$

With the four scene points, we have four distinctive and non-zero form factor terms, *i.e.*, G_{13}, G_{14}, G_{23} and G_{24}. Altogether, there are 5 unknowns, *i.e.*, $\widehat{\mathbf{n}}_1$, $\widehat{\mathbf{n}}_3$, $\widehat{\mathbf{r}}_{12}$, $\widehat{\mathbf{r}}_{13}$ and $\widehat{\mathbf{r}}_{34}$, with 7 degrees of freedom, where $\widehat{\mathbf{r}}_{ij}$ has only 1 degree of freedom as we fix the (x, y) coordinate. Given the 6 equations, the solution space is 1 dimensional. The system is sufficiently constrained if we add another point to either patch, as it introduces 1 additional unknown but 3 more equations. Therefore, in our algorithm, we group 3 points in a patch.

3.4 A Case with Three Patches

As shown in Eq. (12), we can only obtain the form factor up to an unknown albedo value, therefore the actual expression for the form factor term for two mutually visible points $\mathbf{p}_1$ and $\mathbf{p}_3$ is

$$G_{13}(\mathbf{r}_{13}, \widehat{\mathbf{n}}_1, \widehat{\mathbf{n}}_3) = C \frac{(-\widehat{\mathbf{r}}_{13} \cdot \widehat{\mathbf{n}}_1)(\widehat{\mathbf{r}}_{13} \cdot \widehat{\mathbf{n}}_3)}{\|\mathbf{r}_{13}\|^2}, \tag{15}$$

where C is an unknown constant. In the case of uncalibrated projector, the constant C is also needed to account for the unknown scale inherent in the (x, y) coordinate for scene points.

To disambiguate C, we check for the geometric consistency with three patches as shown in Fig. 2 (c). As the solution obtained by evaluating the form factor terms for patch pairs (Π_{12}, Π_{34}) and (Π_{12}, Π_{56}) should agree with that for (Π_{12}, Π_{56}), we can validate the solution of the former with that of the latter. The solutions should best tally when we choose a correct constant C. In the setting of Fig. 2 (c), there are 9 unknowns with 12 degrees of freedom while having 15 equations gives a sufficiently constrained solution space. Without any assumption, the constant C in Eq. (15) is fundamentally unresolvable, as there is a physically feasible surface geometry with a different albedo corresponding to a C. However, with the orthographic assumption mentioned in Sec. 3.1, the constant C is no longer linearly related to depth. With an incorrect C, geometry can be inconsistent in the triangular patch configuration of Fig. 2 (c). Hence, the orthographic assumption breaks the scale ambiguity.

4 Algorithm

As shown in Sec. 3, we need to group at least three scene points into patches in order to obtain a sufficiently constrained system. As another issue, the geometry derived from disjoint point sets will be in different coordinates. In this section, we will look into the criterion for point grouping and the way to bring the geometry at disjoint coordinates into the global coordinate. We will also look at efficient ways for geometry extraction through hierarchical computation or incorporating the prior knowledge of planar surface in the scene.

4.1 Point Grouping

The assumption and guiding principle for point grouping is essentially based on the co-planar property of a point set. An arbitrary set of points is not guaranteed to be co-planar. Hence, we select points that satisfy the following criterion:

- *Adjacency*: The points are adjacent to each other.
- *Mutual invisibility*: Two points with $V_{ij} = 0$ are not mutually visible. $V_{ij} = 0$ implies $G_{ij} = 0$.

Two points satisfying the above criterion are likely to be co-planar. In our implementation, we consider points in a 2×2 neighborhood as being adjacent.

4.2 Pairwise Patch Selection

To make the form factor expression in Eq. (4) more succinct, for three co-planar points $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ sharing a common unit normal vector $\widehat{\mathbf{n}}_1$, we can express $\widehat{\mathbf{n}}_1$ as

$$\widehat{\mathbf{n}}_1 = \frac{\mathbf{r}_{12} \times \mathbf{r}_{23}}{\|\mathbf{r}_{12} \times \mathbf{r}_{23}\|}, \text{ where } \mathbf{r}_{ij} = \mathbf{p}_j - \mathbf{p}_i. \tag{16}$$

If there is an additional point $\mathbf{p}_4$ on the same patch, we need to introduce a constraint to ensure co-planarity:

$$(\mathbf{r}_{12} \times \mathbf{r}_{23}) \cdot \mathbf{r}_{24} = 0. \tag{17}$$

As two points on a patch are having $G_{ij} = 0$ to begin with, we assume that the constraints such as Eq. (17) are automatically satisfied and will not form part of the equations that we are solving. Hence, for two mutually visible patches with N points each, there are $N \times N$ equations for G_{ij} with $2N$ unknowns which correspond to the z-coordinate of the $2N$ points, thus forming a sufficiently constrained system.

In practice, as G_{ij}'s are obtained from measurement, the G_{ij}'s with a low intensity tend to have a low signal to noise ratio and should not be used for computation. As a result, we can have fewer equations while the number of unknowns remains unchanged. To ensure that a patch pair forms a constrained system, we use the following criteria to select a pair of patches with N_a and N_b points respectively:

$$\sum_{i=1}^{N_a} \sum_{j=1}^{N_b} \mathbf{1}(G_{ij} > \epsilon) \geq N_a + N_b \text{ where } \mathbf{1}(\text{true}) = 1 \text{ and } \mathbf{1}(\text{false}) = 0 \tag{18}$$

In our algorithm, we reconstruct the geometry for a pair of patches at a time and then bring the resultant disjoint geometry into the same coordinate frame through the common points connecting the different pairs of patches. However, if there is no direct or indirect visibility link between two points, bringing them into a common coordinate is impossible. The condition for the existence of a global coordinate for all points is that the form factor matrix $\mathbf{G}$ forms a fully connected

Algorithm 1. "Closed-loop" check for reconstructed patches

Require: A list $\mathbf{L}$ of valid patch pairs based on Eq. (18)
1: Sort $\mathbf{L}$ in descending order of the sum of entries in $\mathbf{G}$ for all patch pairs.
2: Start with any patch from the first pair in $\mathbf{L}$ and treat it as a parent patch.
3: Record visited patches into a list $\mathbf{L}_v$. List is blank for the initial patch.
4: **repeat**
5: For a parent patch, identify all its possible branches.
6: Find the best branch based on the sorted list $\mathbf{L}$. If the best branch is in $\mathbf{L}_v$, take the next one.
7: Reconstruct the patch pair and update the depth map and normal map.
8: Push the traversed patch into $\mathbf{L}_v$.
9: Use the traversed patch as parent and Goto 1.
10: **until** The branch patch is the same as the starting patch.
11: Compute the depth error between the initial and final patches.

graph. For every point in a common coordinate frame, we verify its geometry by examining the depth consistency in the closed paths associated to the point. Such closed paths could be many, therefore we only consider the one with the highest intensity and involving at least two other points on different patches. In this process, we are able to identify the reliability of the geometry reconstruction for a point. This consistency check through a "closed-loop" algorithm is presented in Algorithm 1. It is also intended to disambiguate the unknown constant as described in Sec. 3.4.

To increase the reliability of geometry estimation, we perform the above-mentioned steps in a hierarchical manner, from a finer resolution to a coarser one. At one level, we estimate the geometry and group points with similar normals into patches. The patch size grows with increasing level, hence the system of equations for pairwise reconstruction gets more and more constrained and produces more reliable estimation.

Fast method for piece-wise planar scenes: If the scene is known to be piece-wise planar a priori, it is more efficient to adopt a top-down approach for the reconstruction. Except for convex surfaces that do not interact with each other, planar surfaces correspond to "blocks" of zeros. It is worth-noting that the form factor matrix resembles the weight matrix $\mathbf{W}$ in the *Normalized Cut* [21] problem. With this observation, we can segment the scene into planar surface.

5 Experimental Results

To verify our theoretical results, we performed experiments on both synthetic and real data. For synthetic scenes, the reconstruction is based on simulated form factor matrices; while for the real data, the light transport $\mathbf{T}$ of the scene is measured and the form factor matrix $\mathbf{G}$ is derived from $\mathbf{T}$.

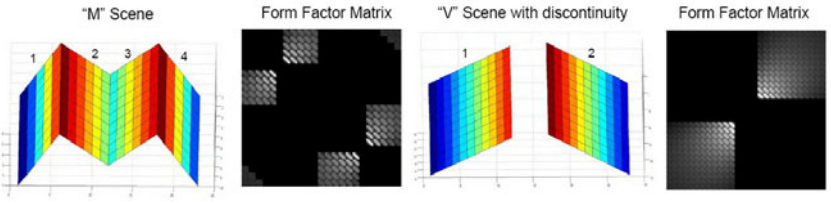


Fig. 3. From left to right: A simulated "M" scene with 11×23 facets and its $\mathbf{G}$ matrix; a simulated inverted "V" scene with discontinuity made of 12×28 facets and its $\mathbf{G}$ matrix. The inner wedge of the "M" scene is made up of 2 convex planes which do not illuminate each other. Note the discontinuity between the 2 planes in the inverted "V" scene. The form factor matrices are log-scaled for display purposes.

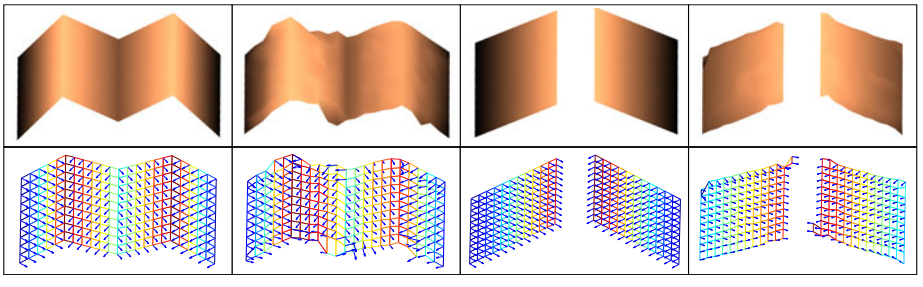


Fig. 4. Top row: Reconstruction results for both clean and noisy simulated "M" and inverted "V" scenes. Bottom row: The recovered surface normal corresponding to the scenes in the top row (normal plotted in opposite directions for display purpose).

5.1 Synthetic Scene

For this experiment, we focus on recovering the shape of simulated 3-D models. To demonstrate the robustness of the proposed method, we perturbed the form factor matrix by additive Gaussian noise. Fig. 3 shows the simulated models and their corresponding form factor matrices.

Fig. 4 shows the reconstruction results for both simulated scenes, using both clean and noisy data. In the noiseless case, perfect recovery of both surface normal and depth can be achieved. Observe that the scale of the reconstruction is the same as the data as we begin with a form factor matrix in the scene's coordinate frame. The recovered structures are subject to a translation in the z-direction as the global depth reference point was arbitrarily set. For the noisy case, both form factor matrices are corrupted by zero-mean Gaussian noise of standard deviation 0.5. In the presence of noise, the shape is better recovered at places such as the joint of 2 planes where the interreflections are stronger. As compared to the recovered depth values, the surface normals are better recovered because they are common among all facet pairs in a particular system of equations. For performance evaluation, we computed the angular error between

Table 1. Shape recovery result. Normal RMSE and angular error between a pair of surfaces are shown. (all errors are measured in degree).

Scene	Normal Angular RMSE	∠ planes 1&2	∠ planes 2&3	∠ planes 3&4
"M" (clean)	0.0018	0.0	0.0	0.0
"M" (noisy)	14.13	5.60	4.30	10.20
inv. "V" (clean)	0.02	0.0	-	-
inv. "V" (noisy)	13.11	6.57	-	-

all estimated surface normals and their ground truth. We also compared the recovered angles between planes with their ground truth in the simulated models. The results are presented in Table. 1.

Handling surface discontinuity: To highlight the proposed method's strength in handling depth discontinuity, we simulated an inverted "V" scene with a gap in the center. Fig. 4 shows the successful reconstruction of this scene. Unlike most shape-from-intensity methods which require the surface to be continuous for the integration of surface orientation, our method is not restricted by surface continuity. Facets lying on occluding boundaries do not have interactions with the rest and therefore do not form any valid equations with them. As a result, these facets will be left unreconstructed since there is insufficient information to determine their relative positions from the others. The same applies to facets lying on the joint between 2 planes. As it is co-planar with both planes, its form factor with facets on both planes equates to 0.

Handling constant factor in G: The constant factor C in Eq. (15) can be determined empirically through closed-loop checks. As we have fixed the x-, y-components of the distance vector and evaluate only the z-component, such a scaling would cause a non-linear change in z. If this factor is not compensated for, the error will show up in the closed-loop check as it propagates through all pair-wise depth estimation before looping back to the starting patch. The error here is defined as the minimum depth error among all close-loop paths. Hence, we can conduct a coarse-to-fine 1-D search to determine the correct factor to cancel C off. To see how C affects the reconstruction, we simulated the recovery of a "V" scene by fitting in different values. The results are presented in Fig. 5. Note that C is being multiplied into the form factors in this experiment but in reality we seek to find a reciprocal to compensate for C. As C deviates from 1, the distortion causes planar surfaces to bend as z changes non-linearly with C.

5.2 Real-World Scene

In this experiment we seek to recover the shape of a real-world scene from its measured light transport matrix. For experimental setup, we used a Canon 5D camera and a Dell 2400MP projector. In our experiment, we consider grayscale light transport for simplicity by assuming that the projector-camera color mixing

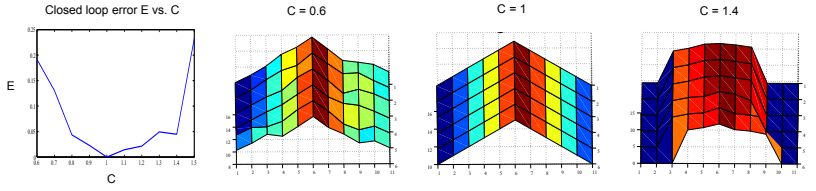


Fig. 5. Top left: A plot of closed-loop error versus C. Second from top left onwards: Reconstruction results by setting C to various values. The recovered shape gets distorted as C deviates from 1.

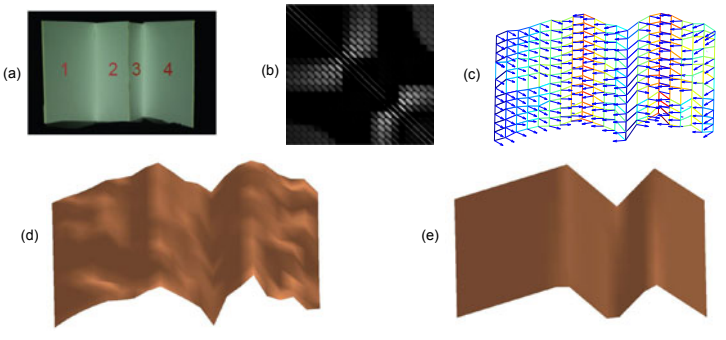


Fig. 6. (a) An image of an "M" scene. (b) The derived form factor matrix **G**. (c) and (d) The recovered surface normal (plotted in opposite direction for display purpose) and shape. Closed loop error is minimized when $C = 0.5$. (e) The recovered shape after plane fitting.

matrix is diagonal. To ensure interreflections is faithfully measured, we used High Dynamic Range capturing with 12 stops of exposures to acquire **T** by a brute-force method. The acquired **T** is verified by bounce separation. In general, the light transport matrix **T** obtained by a projector-camera system has a dimension of $N_c \times N_p$, where N_c and N_p are respectively the number of camera pixels and projector pixels. In this work, we establish a pixel mapping between the camera and the projector by corresponding a camera pixel to the projector pixel that induces a maximum response on it. In our setup, there are more than one camera pixels being mapped to a projector pixel and we group these camera pixels to form a super-pixel. The intensity of a super-pixel is given by the mean of the group of camera pixels. With super-pixels, the resulting **T** matrix takes a square dimension of $N_p \times N_p$. If a super-pixel corresponds to a facet in the scene, with the pixel grouping procedure, we are inherently making uniform-intensity facet assumption.

Fig. 6(a) shows the result of a real "M" scene. The angle between planes 1 and 2 is 80^o and that between planes 3 and 4 is 55^o. (b) shows the derived form factor matrix **G**. For the real data, we first determine the unknown scale factor

C (= 0.5) and multiply $\mathbf{G}$ by $\frac{1}{C}$. Fig. 6(c) and (d) show the recovered normal map and shape; (e) shows the final result after plane fitting. The reconstructed shape is quite close to the original scene. The estimated angle between planes 1 and 2 is 70.43^o and that between planes 3 and 4 is 50.55^o, giving rise to angular errors of 9.57^o and 4.45^o respectively.

6 Conclusion and Discussions

In this paper, we present a method to estimate the scene geometry, *i.e.*, both the depth and the surface normal simultaneously, from a light transport matrix obtained with a projector-camera system. This method can handle a scene with discontinuity. We focused on extracting the geometry information from the second-bounce component that encodes scene interreflections. This method works on convex surface with strong interreflections, which often makes the conventional shape-from-intensity methods fail. Ideally, a complete algorithm for geometry estimation from a light transport matrix should also make use of the first-bounce component, which will help on convex portion of a scene and complement our method. We leave the complete algorithm to future work. Light transport is often applied for relighting applications that assume static light transport. The capability to estimate geometry will open up opportunities in fast acquisition of dynamic-scene light transport and make light transport editing possible for graphics applications. In future, we will look into more robust signal processing techniques to improve the shape reconstruction.

Limitations. One limitation of the proposed reconstruction algorithm lies in the concavity of the scene. Standalone convex surface cannot be reconstructed. However, if there exist other surfaces in the scene forming concave pairs with it, the geometry of this locally convex surface can still be recovered, *e.g.*, the inner wedge of the "M" scene can be reconstructed despite its convex nature, as the 2 inner planes interact with the outer planes to form concave pairs.

References

1. Nayar, S., Ikeuchi, K., Kanade, T.: Shape from interreflections. In: Proc. of Int'l Conf. on Computer Vision, pp. 2–11 (1990)
2. Yu, Y., Debevec, P., Malik, J., Hawkins, T.: Inverse global illumination: recovering reflectance models of real scenes from photographs. In: Proc. of ACM SIGGRAPH, pp. 215–224 (1999)
3. Machida, T., Yokoya, N., Takemura, H.: Surface reflectance modeling of real objects with interreflections. In: Proc. of Int'l Conf. on Computer Vision, pp. 170–177 (2003)
4. Seitz, S.M., Matsushita, Y., Kutulakos, K.N.: A theory of inverse light transport. In: Proc. of Int'l Conf. on Computer Vision, pp. 1440–1447 (2005)
5. Nayar, S.K., Krishnan, G., Grossberg, M.D., Raskar, R.: Fast separation of direct and global components of a scene using high frequency illumination. In: Proc. of ACM SIGGRAPH, pp. 935–944 (2006)

6. Gupta, M., Tian, Y., Narasimhan, S.G., Zhang, L.: (de) focusing on global light transport for active scene recovery. In: Proc. of Computer Vision and Pattern Recognition (2009)
7. Ng, T.T., Pahwa, R.S., Tan, K.H., Bai, J., Quek, T.Q.S.: Radiometric compensation using stratified inverses. In: Proc. of Int'l Conf. on Computer Vision (2009)
8. Foley, J.D., van Dam, A., Feiner, S.K., Hughes, J.F.: Computer Graphics: Principles and Practice (1990)
9. Goral, C.M., Torrance, K.E., Greenberg, D.P., Battaile, B.: Modelling the interaction of light between diffuse surfaces. Computre Graphics 18, 212–222 (1984)
10. Debevec, P., Hawkins, T., Tchou, C., Duiker, H.P., Sarokin, W., Sagar, M.: Acquiring the reflectance field of a human face. In: Proc. of ACM SIGGRAPH, pp. 145–156 (2000)
11. Goesele, M., Lensch, H.P., Lang, J., Fuchs, C., Seidel, H.P.: Disco: acquisition of translucent objects. In: Proc. of ACM SIGGRAPH, pp. 835–844 (2004)
12. Sen, P., Chen, B., Garg, G., Marschner, S., Horowitz, M., Levoy, M., Lensch, H.: Dual photography. In: Proc. of ACM SIGGRAPH, pp. 745–755 (2005)
13. Peers, P., Mahajan, D.K., Lamond, B., Ghosh, A., Matusik, W., Ramamoorthi, R., Debevec, P.: Compressive light transport sensing. ACM Trans. Graph. 28, 1–18 (2009)
14. Yu, Y., Debevec, P., Malik, J., Hawkins, T.: Inverse global illumination: Recovering reflectance models of real scenes from photographs. In: Proc. of ACM SIGGRAPH, pp. 215–224 (1999)
15. Boyer, K.L., Kak, A.C.: Color-coded structured light for rapid active ranging. IEEE Trans. on Pattern Analysis and Machine Intelligence 9, 14–28 (1987)
16. Harley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn (2004)
17. Lambert, J.H.: Photometry, or, on the measure and gradations of light, colors, and shade: Translation from the latin of photometria, sive, de mensura et gradibus luminis, colorum et umbrae, 1760. Illuminating Eng. Soc. of North America (2001)
18. Schröder, P., Hanrahan, P.: On the form factor between two polygons. In: Proc. of ACM SIGGRAPH, pp. 163–164 (1993)
19. Kajiya, J.T.: The rendering equation. In: Proc. of ACM SIGGRAPH, pp. 143–150 (1986)
20. Arvo, J., Torrance, K., Smiths, B.: A framework for the analysis of error in global illumination algorithms. In: Proc. of ACM SIGGRAPH, pp. 75–84 (1994)
21. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence 22, 888–905 (2000)

A Dual Theory of Inverse and Forward Light Transport

Jiamin Bai[1], Manmohan Chandraker[1],
Tian-Tsong Ng[2], and Ravi Ramamoorthi[1]

[1] University of California, Berkeley
[2] Institute for Infocomm Research, Singapore

Abstract. Inverse light transport seeks to undo global illumination effects, such as interreflections, that pervade images of most scenes. This paper presents the theoretical and computational foundations for inverse light transport as a dual of forward rendering. Mathematically, this duality is established through the existence of underlying Neumann series expansions. Physically, we show that each term of our inverse series cancels an interreflection bounce, just as the forward series adds them. While the convergence properties of the forward series are well-known, we show that the oscillatory convergence of the inverse series leads to more interesting conditions on material reflectance. Conceptually, the inverse problem requires the inversion of a large transport matrix, which is impractical for realistic resolutions. A natural consequence of our theoretical framework is a suite of fast computational algorithms for light transport inversion – analogous to finite element radiosity, Monte Carlo and wavelet-based methods in forward rendering – that rely at most on matrix-vector multiplications. We demonstrate two practical applications, namely, separation of individual bounces of the light transport and fast projector radiometric compensation to display images free of global illumination artifacts in real-world environments.

1 Introduction

Global illumination effects are key visual features of real-world scenes. Simulation of these effects in forward rendering has been extensively studied in computer graphics, with a theoretical foundation based on the rendering equation [9]. In contrast, most computer vision algorithms are forced to simply ignore interreflections, where one would ideally like to invert the rendering equation to undo their effects. Recently, Seitz et al. [18] formalized this as the problem of inverse light transport. However, little is known about the theory and algorithms for efficient light transport inversion in practical scenes.

This paper lays the mathematical and computational foundations of inverse light transport, by exposing a strong duality to the already mature framework of forward light transport. Intuitively, the duality arises because solving the (forward) rendering equation itself involves an operator or matrix inverse. Exploiting this duality allows us to leverage many theoretical results and algorithms from forward global illumination for the inverse problem in computer vision.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 294–307, 2010.

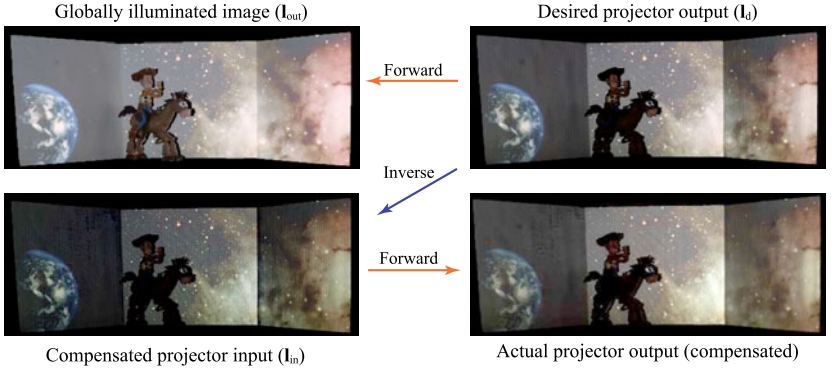


Fig. 1. Application of inverse light transport for projector compensation in a real scene. **Top:** The desired projector output (right) leads to significant interreflections when displayed (left). **Bottom:** Our theory determines the pattern (left) whose projection is close to the desired (right). Our fast iterative method involves only matrix-vector multiplications, with each iteration taking only 0.03 sec. For a transport matrix of size $10^5 \times 10^5$, the full image $\mathbf{l}_{\text{in}}$ is computed after several iterations in 2-3 secs.

Specifically, forward rendering readily admits to a Neumann series solution. We derive a similar series for the inverse solution and show formally that just as each term of the forward Neumann series adds bounces of light transport, each term of the inverse series zeroes out the corresponding bounce (but unlike in the forward case, also affects higher-order bounces). However, convergence of the inverse series is oscillatory. While the forward series convergence condition corresponds to energy conservation, in the inverse case the condition is more complex—a sufficient condition is that the albedo of surfaces is below 0.5, so that the net global illumination is still less than the direct lighting component.

Recent techniques for acquiring the light transport of real scenes [11,16] have facilitated relighting applications in computer graphics, equivalent to matrix-vector multiplication. While light transport *inversion* enables new applications like illumination estimation, separating bounces of global illumination [18] and projector radiometric compensation [21], the high resolution of real transport data ($10^5 \times 10^5$ or higher) often makes standard matrix inversion impractical.

Inspired by efficient algorithmic approaches such as finite element radiosity [4] and Monte Carlo methods [9,20] for the forward problem, we propose fast algorithms for canceling interreflections, which require only matrix-vector multiplications (as opposed to a full matrix inversion). We demonstrate practical applications of these algorithms, such as full radiometric compensation of interreflections while projecting complex scenes (Fig. 1), as well as separation of individual local and global illumination components or bounces (Fig. 2).

To summarize, the main contributions of this paper are:

– A theoretical framework that provides novel insights into light transport inversion by posing it as a dual to forward rendering.

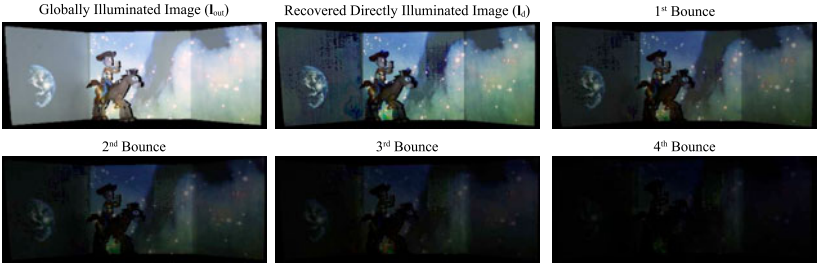


Fig. 2. Separation of bounces of interreflection using our iterative light transport inversion technique, that runs in 3 seconds on a 131K × 131K light transport matrix

- Efficient algorithms for inverting high resolution light transport, with rigorous convergence and error analysis.
- Demonstration of practical applications such as bounce separation and radiometric compensation in complex, non-Lambertian scenes.

2 Previous Work

Our work builds most closely on Seitz et al. [18], who introduce the problem of inverse light transport. This paper elucidates novel theoretical connections to the forward problem and proposes new algorithms that are far more efficient (hence, practical on high resolution data) than the direct matrix inversion of [18]. Nayar et al. [13] present a fast direct and global separation where the entire scene is lit by a light source. In contrast, we acquire the full light transport, but can then separate each bounce of light and consider general illumination conditions.

Our approach is distinct from inverse rendering methods [10,17] that acquire lighting and reflectance, as well as the inverse global illumination method of [22] for BRDF estimation, all of which assume known scene geometry. In contrast, we observe only the light transport matrix—both geometry and reflectance are unknowns in this work. We focus on the case where scene elements are illuminated individually by a single projector, with a camera recording the output [15,16,18]. Extensions to incident (and reflected) light fields [7,11,19] are encompassed by the theory, but not yet considered in our practical applications.

One important application is projector radiometric compensation, where we seek to project a desired image, while compensating for global illumination effects on the display surface. Many recent efforts considered non-uniform scene reflectance, but not interreflections [5,6,14]. Clusters of camera-projector pixels are formed in [21], with a brute-force transport inversion within clusters, but inter-cluster interactions are ignored. Iterative inverse methods for diffuse scenes are proposed in [3], and a series expansion for inverse light transport, denoted as the stratified inverse, is derived by Ng et al. [15]. We show that this series is a natural analog to the forward Neumann series. Our dual formulation enables us to go much further, by showing that the inverse series subtracts

$\mathbf{l}_{\text{in}}$	Incident lighting or projected pattern	$\mathbf{l}_{\text{d}}$	Direct light from sources
$\mathbf{l}_g$	Global light from interreflections	$\mathbf{l}_{\text{out}}$	Outgoing light ($\mathbf{l}_{\text{out}} = \mathbf{l}_{\text{d}} + \mathbf{l}_g$)
$\mathbf{S}$	Forward transport operator	$\mathbf{S}^{-1}$	Inverse transport operator
$\mathbf{R}$	Interreflections operator ($\mathbf{R} = \mathbf{S} - \mathbf{I}$)	$\mathbf{K}$	Local reflection operator
$\mathbf{G}$	Geometric operator	$\mathbf{A}$	Net global transport, $\mathbf{A} = \mathbf{KG}$
$\mathbf{F}$	First bounce from projector	$\mathbf{T}$	Observed light transport, $\mathbf{T} = \mathbf{SF}$
N	Transport resolution (matrix size N^2)	p	$\|\mathbf{K}\|$, related to max albedo ($p < 1$)

Fig. 3. Table of the main notation used in the paper

physical bounces of light, analyzing convergence conditions and providing fast algorithms that relate to radiosity, wavelet and Monte Carlo methods in forward rendering.

3 Preliminaries

Owing to the linearity of light transport, the image formation process is governed by a linear operator $\mathbf{S}$ that encodes the effects of global illumination:

$$\mathbf{l}_{\text{out}} = \mathbf{S}\mathbf{l}_{\text{d}}, \tag{1}$$

where $\mathbf{l}_{\text{out}}$ is the outgoing "global" light, and $\mathbf{l}_{\text{d}}$ is the direct lighting on surfaces due to external sources. In continuous form, $\mathbf{l}_{\text{out}}$ and $\mathbf{l}_{\text{d}}$ are functions (of spatial location and outgoing direction), while $\mathbf{S}$ is a linear operator that accounts for global illumination. When discretized for practical applications, $\mathbf{l}_{\text{out}}$ and $\mathbf{l}_{\text{d}}$ are vectors, while $\mathbf{S}$ is the interreflection matrix. Note that (1) depends only on linearity, and holds for the light field, as well as a single camera view (image).

Unlike forward global illumination computations, we do not see the light source directly, but rather its effect on the scene, which we denote as the direct component, $\mathbf{l}_{\text{d}}$. The inverse light transport problem considered here is simply

$$\mathbf{l}_{\text{d}} = \mathbf{S}^{-1}\mathbf{l}_{\text{out}}, \tag{2}$$

where we seek to invert the operator $\mathbf{S}^{-1}$, undoing the effects of interreflections.

Practical Issues: In practice, it is rare that $\mathbf{S}$ is measured directly. Instead, a projector or illumination source lights the scene,

$$\mathbf{l}_{\text{d}} = \mathbf{F}\mathbf{l}_{\text{in}} \qquad \mathbf{l}_{\text{out}} = \mathbf{T}\mathbf{l}_{\text{in}} = \mathbf{SF}\mathbf{l}_{\text{in}}, \tag{3}$$

where $\mathbf{l}_{\text{in}}$ is the incident pattern projected (or distant light sources turned on), and $\mathbf{F}$ is a "first-bounce" matrix or operator. The actual acquired light transport is $\mathbf{T} = \mathbf{SF}$. The above expression holds for any light transport acquisition system.

The remainder of the theoretical development focuses on analyzing and computing $\mathbf{S}^{-1}$. Eventual practical applications do need to convert from $\mathbf{T}$ to $\mathbf{S}$, using $\mathbf{S} = \mathbf{T}\mathbf{F}^{-1}$. Moreover, applications like radiometric compensation actually seek to recover $\mathbf{l}_{\text{in}}$ (rather than $\mathbf{l}_{\text{d}}$ in (2)) given by $\mathbf{l}_{\text{in}} = \mathbf{T}^{-1}\mathbf{l}_{\text{out}} = \mathbf{F}^{-1}\mathbf{l}_{\text{d}}$.

Since we focus on global illumination $\mathbf{S}$, we will consider setups where $\mathbf{S}$ is easy to obtain from $\mathbf{T}$, i.e., where $\mathbf{F}$ is simple and at least approximately invertible. Therefore, we consider projector-based acquisition, that illuminates a single spatial location, rather than light sources that illuminate the whole object (where $\mathbf{F}$ is low rank for diffuse surfaces [17]). After geometric calibration, we can use the same parameterization for projection and camera images [18]. $\mathbf{F}$ is then a diagonal matrix, with $\mathbf{F}^{-1}$ being trivial to compute.

Note that $\mathbf{F}$ need not correspond to the actual first bounce for an accurate light transport inversion. In numerical terms, choosing $\mathbf{F} = \mathrm{diag}(\mathbf{T})$ amounts to Jacobi preconditioning, which is convergent if $\mathbf{T}$ is diagonally dominant.

4 Dual Forward and Inverse Light Transport

In this section, we show that the structure of the rendering equation exposes a strong duality between forward and inverse light transport. We derive analogous inverse and forward Neumann series expansions, and interpret them in terms of physical bounces of light. Key theoretical results are summarized in Fig. 4.

	Forward	Inverse
Problem	$\mathbf{l}_{\text{out}} = \mathbf{S}\mathbf{l}_{\text{d}}$	$\mathbf{l}_{\text{d}} = \mathbf{S}^{-1}\mathbf{l}_{\text{out}}$
Duality	$\mathbf{S} = (\mathbf{I} - \mathbf{A})^{-1}$	$\mathbf{S}^{-1} = (\mathbf{I} + \mathbf{R})^{-1}$
Series	$\mathbf{S} = \mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \dots$	$\mathbf{S}^{-1} = \mathbf{I} - \mathbf{R} + \mathbf{R}^2 - \dots$
Bounces	$\mathbf{S}_n = \sum_{k=0}^{n} \mathbf{A}^k = \mathbf{S} + O(\mathbf{A}^{n+1})$	$\mathbf{S}_n^{-1} = \mathbf{S}^{-1} + O(\mathbf{A}^{n+1})$
Iteration	$\mathbf{l}_{\text{out}}^{(k)} = \mathbf{l}_{\text{d}} + \mathbf{A}\mathbf{l}_{\text{out}}^{(k-1)}$	$\mathbf{l}_{\text{d}}^{(k)} = \mathbf{l}_{\text{out}} - \mathbf{R}\mathbf{l}_{\text{d}}^{(k-1)}$
Monte Carlo	$\sum \mathbf{A}_{i_0 i_1} \mathbf{A}_{i_1 i_2} \dots \mathbf{l}_{\text{d}}(i_k)$	$\sum (-1)^k \mathbf{R}_{i_0 i_1} \mathbf{R}_{i_1 i_2} \dots \mathbf{l}_{\text{out}}(i_k)$

Fig. 4. Duality of forward and inverse light transport, indicating analogous relations for some key properties. (Monte Carlo equations abbreviated; full forms in text).

In the operator notation of [1], the rendering equation is written as

$$\mathbf{l}_{\text{out}} = \mathbf{l}_{\text{d}} + \mathbf{K}\mathbf{G}\mathbf{l}_{\text{out}} \quad \Rightarrow \quad \mathbf{l}_{\text{out}} = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{l}_{\text{d}}, \tag{4}$$

where $\mathbf{K}$ considers the local reflection at a surface, governed by the BRDF, $\mathbf{G}$ is a geometric operator that transports outgoing to incident radiance and $\mathbf{A} = \mathbf{K}\mathbf{G}$ corresponds to one physical bounce of light.[1] It naturally follows that

$$\boxed{\mathbf{S} = (\mathbf{I} - \mathbf{A})^{-1}.} \tag{5}$$

[1] This formulation is valid for any opaque BRDF when considering the full light field. While the theory is fully general, our experiments will consider projection to a single view, which introduces practical limitations, as discussed in Sec. 7.

This well known result shows that the forward problem formally involves a matrix or operator inversion. Also note that if the scene geometry and reflectance (and hence $\mathbf{A}$) are known, we simply have $\mathbf{S}^{-1} = \mathbf{I} - \mathbf{A}$, as noted by [18,12]. We focus here on cases where we only measure $\mathbf{S}$, but do not know or compute $\mathbf{A}$.

We can separate $\mathbf{l}_{\text{out}}$ into direct $\mathbf{l}_{\text{d}}$ and indirect or global $\mathbf{l}_g$ components,

$$\mathbf{l}_{\text{out}} = \mathbf{l}_{\text{d}} + \mathbf{l}_g = \mathbf{l}_{\text{d}} + \mathbf{R}\mathbf{l}_{\text{d}} \qquad \mathbf{l}_{\text{out}} = (\mathbf{I} + \mathbf{R})\mathbf{l}_{\text{d}} \tag{6}$$

where $\mathbf{R} = \mathbf{S} - \mathbf{I}$ is a linear operator that accounts only for global illumination. We are now ready to present an expression for inverse light transport:

$$\boxed{\mathbf{S}^{-1} = (\mathbf{I} + \mathbf{R})^{-1}.} \tag{7}$$

The very similar or dual forms of (5) and (7) is a key insight in this paper, and allows direct leveraging of many forward rendering theories and algorithms for inverse rendering and inverse light transport algorithms in computer vision.

Neumann Forward and Inverse Series: The forward equations (4) and (5) have well-known series expansions corresponding physically to light bounces,

$$\boxed{\mathbf{S} = \mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \ldots.} \tag{8}$$

We can also relate the global illumination operator $\mathbf{R}$ to this expansion,

$$\mathbf{R} = \mathbf{S} - \mathbf{I} = \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \ldots. \tag{9}$$

Mathematically, the dual formulation in (7) has a series analogous to (8),

$$\boxed{\mathbf{S}^{-1} = \mathbf{I} - \mathbf{R} + \mathbf{R}^2 - \mathbf{R}^3 + \ldots.} \tag{10}$$

Note that the positive sign of $\mathbf{R}$ implies the series is oscillatory. Intuitively, from (6), $\mathbf{l}_{\text{d}} = \mathbf{l}_{\text{out}} - \mathbf{R}\mathbf{l}_{\text{d}}$. Since the unknown $\mathbf{l}_{\text{d}}$ appears on the right hand side, a first approximation as $\mathbf{l}_{\text{d}} \approx \mathbf{l}_{\text{out}}$ calculates $\mathbf{l}_{\text{d}} \approx \mathbf{l}_{\text{out}} - \mathbf{R}\mathbf{l}_{\text{out}}$. This overcompensation is corrected by higher-order terms, leading to the alternating signs in (10).

With suitable algebraic manipulations, one may note that (10) explains the stratified inverses of Ng *et al.* [15] and relates it to the rendering equation.[2]

Interpretation as Physical Bounces of Light: Consider an approximation up to order n, that we denote as $\mathbf{S}_n$ or $\mathbf{S}_n^{-1}$. In the forward case, it is clear that

$$\boxed{\mathbf{S}_n = \sum_{k=0}^{n} \mathbf{A}^k \qquad \mathbf{S}_n - \mathbf{S} = O(\mathbf{A}^{n+1})} \tag{11}$$

where the first n physical bounces of light are represented (each term adds the next bounce), and the error is from neglecting bounces $n+1$ onwards.

[2] In particular, note that $\mathbf{R} = \mathbf{S} - \mathbf{I}$, which is $\mathbf{T}\mathbf{F}^{-1} - \mathbf{I}$. A final binomial expansion in $\mathbf{T}\mathbf{F}^{-1}$ and using $\mathbf{T}^{-1} = \mathbf{F}^{-1}\mathbf{S}^{-1}$ enables one to derive the results in [15].

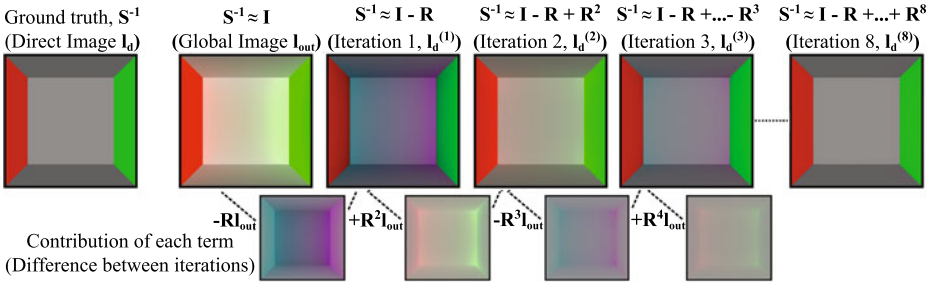


Fig. 5. Top: From left to right, we add more terms of the inverse series, going from the simulated global illumination output $\mathbf{l}_{\text{out}}$ to the "direct lighting" result $\mathbf{l}_{\text{d}}$ (shown leftmost). These terms also correspond to the iterations introduced in Sec. 6. **Bottom:** Contributions of individual terms (neutral grey is 0).

A physical interpretation for the inverse series seems non-intuitive, since (10) is expressed in terms of $\mathbf{R}$, that includes all global illumination terms. Nevertheless, in [2], we derive a surprising result: each term of the inverse series *cancels or zeros out the corresponding bounce of light transport*, analogous to the forward case. Formally, we show that $\mathbf{S}_0^{-1} = \mathbf{I}$, $\mathbf{S}_1^{-1} = \mathbf{I} - \mathbf{A} - \mathbf{A}^2 - \ldots$, and for $n > 1$,

$$\mathbf{S}_n^{-1} = \mathbf{I} - \mathbf{A} + \sum_{m=2}^{\infty} \left[\sum_{k=1}^{\min(m,n)} (-1)^k \binom{m-1}{k-1} \right] \mathbf{A}^m. \tag{12}$$

Now, consider the case when $m \leq n$. In this case, the second summation has a limit of $m > 1$, and the coefficient of $\mathbf{A}^m$ becomes $\sum_{k=1}^{m} (-1)^k \binom{m-1}{k-1}$, which is the binomial expansion of $(1+x)^{m-1}$ with $x = -1$, thus, identically 0.

This implies a key result, that the $\mathbf{A}^m$ terms vanish for $2 \leq m \leq n$,

$$\boxed{\mathbf{S}_n^{-1} = \mathbf{I} - \mathbf{A} + O(\mathbf{A}^{n+1}) \qquad \mathbf{S}_n^{-1} - \mathbf{S}^{-1} = O(\mathbf{A}^{n+1})} \tag{13}$$

analogous to the forward series in (11). An exact expression can be derived as $\mathbf{S}_n^{-1} = \mathbf{I} - \mathbf{A} + (-1)^n \sum_{m=n+1}^{\infty} \binom{m-2}{n-1} \mathbf{A}^m$. Note the oscillatory series behavior from the $(-1)^n$. Finally, since $\mathbf{S} = (\mathbf{I} - \mathbf{A})^{-1}$,

$$\mathbf{S}_n^{-1}\mathbf{S} = \left[(\mathbf{I} - \mathbf{A}) + O(\mathbf{A}^{n+1}) \right] [\mathbf{I} - \mathbf{A}]^{-1} = \mathbf{I} + O(\mathbf{A}^{n+1}). \tag{14}$$

In other words, the n term series $\mathbf{S}_n^{-1}$ annihilates the first n physical bounces of light (each term in the series zeroes the corresponding interreflection bounce), leaving only bounces $n+1$ and higher. However, as opposed to the forward series where the higher bounces are simply 0 until they are added in, the values for the higher bounces in the inverse series oscillate until they are zeroed—this is related to the oscillatory convergence of the inverse series. An exact result is

$$\mathbf{S}_n^{-1}\mathbf{S} = \mathbf{I} + \quad (-1)^n \sum_{m=n+1}^{\infty} \binom{m-1}{n} \mathbf{A}^m. \tag{15}$$

5 Convergence and Error Analysis

For the forward case, Arvo et al. [1] prove several results, briefly summarized here. For a closed enclosure, $\|\mathbf{G}\| = 1$ (less for open scenes). From energy conservation, excluding perfect reflectors, $\|\mathbf{K}\| \leq p < 1$, where p relates to the surface albedo (for non-diffuse materials, it is the maximum over all incident directions of the fraction of total energy reflected).[3] Since $\|\mathbf{A}\| \leq \|\mathbf{K}\|\|\mathbf{G}\|$, it follows that $\|\mathbf{A}\| \leq p < 1$, so the forward series always converges.

For the inverse series in (10), a bound from (9) is,

$$\|\mathbf{R}\| \leq \|\mathbf{A}\| + \|\mathbf{A}^2\| + \ldots \leq p + p^2 + \ldots = \frac{p}{1-p}. \tag{16}$$

If $p < \frac{1}{2}$, we obtain $\|\mathbf{R}\| < 1$, which is sufficient for convergence (though *not* necessary). Intuitively, if the diffuse albedo (or maximum fraction of energy reflected for any incident direction for non-diffuse materials) is less than 1/2, the norm of the total global illumination operator $\mathbf{R}$ is less than that of the direct lighting operator $\mathbf{I}$. In matrix terms, $\mathbf{S} = \mathbf{I} + \mathbf{R}$ is diagonally dominant. Since the inverse series is oscillatory, we require to bound the full global illumination, rather than just each bounce separately as in the forward case.

Error Analysis: The error introduced in an n term expansion ($\mathbf{S}_n$ or $\mathbf{S}_n^{-1}$) for forward and inverse series can be bounded as

$$\|\mathbf{S} - \mathbf{S}_n\| \leq \sum_{k=n+1}^{\infty} \|\mathbf{A}^k\| \leq \sum_{k=n+1}^{\infty} p^k = \frac{p^{n+1}}{1-p}. \tag{17}$$

$$\|\mathbf{S}^{-1} - \mathbf{S}_n^{-1}\| \leq \sum_{k=n+1}^{\infty} \|\mathbf{R}^k\| \leq \sum_{k=n+1}^{\infty} \left(\frac{p}{1-p}\right)^k = \frac{p^{n+1}}{(1-p)^n(1-2p)}. \tag{18}$$

Numerical Simulations: For simplicity, we consider a synthetic diffuse box (closed, so $\|\mathbf{G}\| = 1$), without shadows but with interreflections. Fig. 5 assumes that $\mathbf{l}_\mathrm{d}$ is constant on each surface, which have different albedos. From left to right, addition of more terms from (10) causes oscillations between over and under-compensating interreflections, till convergence to $\mathbf{l}_\mathrm{d}$. Interestingly, while forward global illumination in $\mathbf{l}_\mathrm{out}$ results in predictable red and green color-bleeding, odd terms of the inverse series give rise to cyan and magenta colors. The final inverse light transport solution for $\mathbf{l}_\mathrm{d}$ has no color bleeding, as desired.

In Fig. 6, we analyze errors and convergence. Fig. 6a indicates similar oscillatory convergence behavior near corners, edges and face centers. Fig. 6b shows

[3] These relations hold in any L^p norm, since by reciprocity, $\|\mathbf{K}\|_1 = \|\mathbf{K}\|_\infty = p$, and $\|\cdot\|_q \leq \max(\|\cdot\|_1, \|\cdot\|_\infty)$.

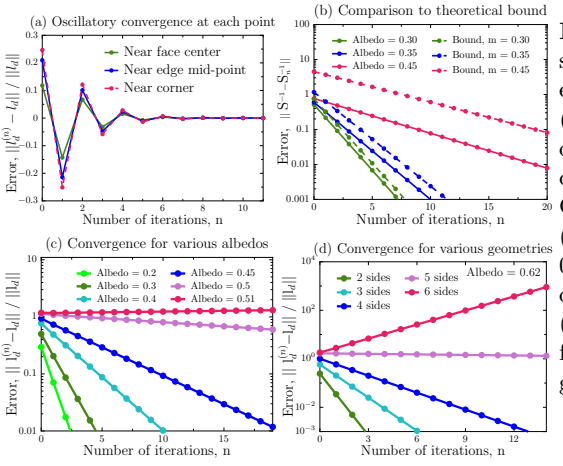


Fig. 6. Error analysis of inverse series. **(a):** Convergence at different points (center, edge, corner). **(b):** Comparison of error to theoretical bound for different albedos showing good agreement. **(c):** Convergence for different albedos (is faster for lower albedos up to 0.5). **(d):** As expected, an albedo of 0.62 diverges for a closed box (6 sides), shows slow convergence for a 5-sided box and rapid convergence for more open environments.

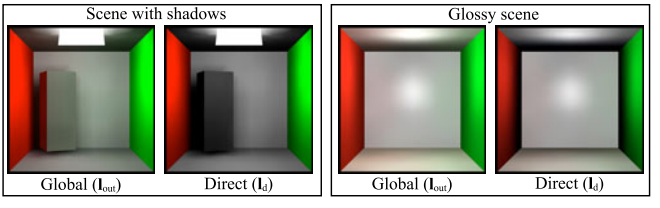


Fig. 7. Validation of the theory for shadowed and non-Lambertian scenes. Our iterative method recovers $\mathbf{l}_d$ in 10 iterations for the shadowed scene and 20 for the glossy one.

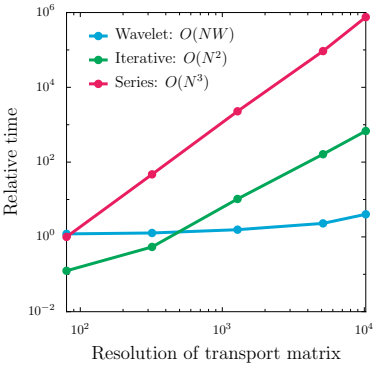
excellent agreement, up to a constant factor, between error for the whole $\mathbf{S}^{-1}$ operator and the theoretical bound in (18). Fig. 6c illustrates the inverse relation of convergence rate and albedo. Even albedos near the theoretical limit (like 0.45) converge in a few iterations, those very close to 0.5 converge slowly and those greater than 0.51 diverge. Fig. 6d shows the variation of convergence with geometry (that is, $\|\mathbf{G}\|$). For an albedo of 0.62, close to the theoretical limit for a 5-sided box, we observe very slow convergence for a 5-sided box, divergence for a 6-sided box and rapid convergence for more open geometries.

Finally, Fig. 7 shows a scene with occlusions and glossy surfaces. Similar behaviors hold as above, with convergence of the inverse series to direct lighting.

6 Exploiting Duality for Fast Light Transport Inversion

We now introduce efficient algorithms for high-resolution light transport inversion, exploring duals to iterative finite element radiosity, wavelet accelerations and Monte Carlo methods.

Finite Element Methods: Forward rendering rarely computes the series in (8) to explicitly determine $\mathbf{S}$, due to the high cost of matrix-matrix multiplications on high-resolution scenes. Instead, finite element and radiosity methods [4] try to solve $\mathbf{l}_{out} = \mathbf{l}_d + \mathbf{A}\mathbf{l}_{out}$, which corresponds directly to (4), iteratively,



Method	Resolution of transport matrix (N)				
	80	320	1280	5120	10240
Series	1.0	47.0	2.3e3	9.3e4	7.5e5
Iterative	0.1	0.5	10.3	162.5	679.0
Wavelet	1.2	1.3	1.6	2.3	4.0

Fig. 8. Timings for series, iterative finite element, and wavelet accelerated methods (using Daubechies4 wavelets). N is the transport resolution (matrix size is N^2). We normalize timings so that 1.0 corresponds to 5.57×10^{-4} seconds, with experiments in Matlab on an Intel i7 machine. All methods are run until 1% error.

$$\boxed{\mathbf{l}_{\text{out}}^{(k)} = \mathbf{l}_{\text{d}} + \mathbf{A}\mathbf{l}_{\text{out}}^{(k-1)}.} \tag{19}$$

This iteration is numerically stable, and requires only the matrix-vector multiplication for $\mathbf{A}\mathbf{l}_{\text{out}}$. The superscript stands for the step k, and $\mathbf{l}_{\text{out}}^{(0)} = \mathbf{l}_{\text{d}}$. Note that n steps simply compute the effect of the first n terms of the series in (8). For inverse light transport, one can derive a similar relation, starting from $\mathbf{l}_{\text{d}} = \mathbf{l}_{\text{out}} - \mathbf{R}\mathbf{l}_{\text{d}}$, that follows from (6). The iterative solution naturally follows, dual to (19),

$$\boxed{\mathbf{l}_{\text{d}}^{(k)} = \mathbf{l}_{\text{out}} - \mathbf{R}\mathbf{l}_{\text{d}}^{(k-1)},} \tag{20}$$

with $\mathbf{l}_{\text{d}}^{(0)} = \mathbf{l}_{\text{out}}$. Again, the first n steps correspond to the first n terms in (10). Note the negative sign on $\mathbf{R}$ that determines the oscillatory nature of the series.

Matrix Iteration: In cases where we seek to precompute $\mathbf{S}^{-1}$, there is also a corresponding full matrix iteration. The dual forward and inverse relations are

$$\mathbf{S}_k = \mathbf{I} + \mathbf{A}\mathbf{S}_{k-1}, \qquad \mathbf{S}_k^{-1} = \mathbf{I} - \mathbf{R}\mathbf{S}_{k-1}^{-1}, \tag{21}$$

with $\mathbf{S}_0 = \mathbf{S}_0^{-1} = \mathbf{I}$. These equations provide a numerically stable iteration.

Wavelet Methods: The matrix-vector multiplication $\mathbf{R}\mathbf{l}_{\text{d}}$ in (20) is the time-consuming step. We can wavelet-transform and approximate the vector $\mathbf{l}_{\text{d}}$, as well as the rows of $\mathbf{R}$, to speed it up. This is analogous to wavelet radiosity and light transport in forward rendering [8].

Monte Carlo Methods: For the matrix $\mathbf{A}$, [9] considers all index permutations

$$\mathbf{l}_{\text{out}}(i_0) = \mathbf{l}_{\text{d}}(i_0) + \sum_{k=1}^{\infty} \sum_{i_1, i_2, \ldots i_k} \mathbf{A}_{i_0 i_1} \mathbf{A}_{i_1 i_2} \ldots \mathbf{A}_{i_{k-1} i_k} \mathbf{l}_{\text{d}}(i_k), \tag{22}$$

where the first summation is over all terms k in the series, or all path lengths in a path tracing context. The different indices correspond to all matrix sums, or equivalently all paths, where each i_j chooses a particular point on the path.

Analogously, the inverse series in (10) has a similar form,

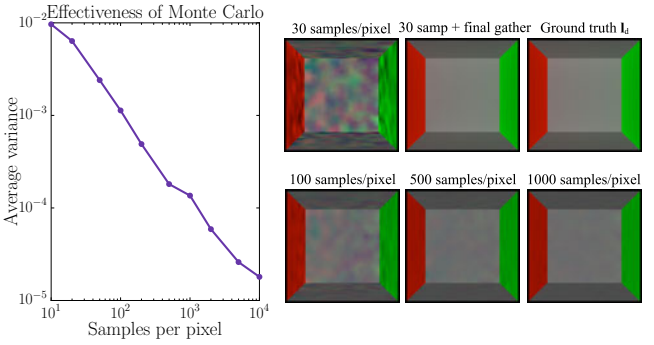


Fig. 9. Left: Variance in Monte Carlo methods. **Right:** Top row shows final gather inversion with only 30 samples. Bottom row shows effects of increasing samples with pure Monte Carlo. Transport resolution is $N = 5120$.

$$\mathbf{l}_{\mathrm{d}}(i_0) = \mathbf{l}_{\mathrm{out}}(i_0) + \sum_{k=1}^{\infty} (-1)^k \sum_{i_1, i_2, \ldots i_k} \mathbf{R}_{i_0 i_1} \mathbf{R}_{i_1 i_2} \ldots \mathbf{R}_{i_{k-1} i_k} \mathbf{l}_{\mathrm{out}}(i_k), \tag{23}$$

where the oscillatory behavior requires the additional $(-1)^k$ factor. A direct Monte Carlo algorithm uses a number of samples, drawing the indices $i_1, i_2, \ldots i_k$ at random for each. The expectation of these samples gives the desired result. One may also use fewer samples for the iteration, but compute the final step with a direct matrix-vector multiplication, akin to *final gather* in forward rendering.

Numerical Simulations: As timing baseline, we use matrix-matrix multiplications to directly compute the series in (10) (explicit matrix inversion is intractable for high resolutions). In Fig. 6, for transport resolution N, the series method scales as $O(N^3)$ and rapidly becomes impractical. The iterative method uses only matrix-vector multiplications and is $O(N^2)$, with a speedup of three orders of magnitude for large sizes. Wavelet acceleration leads to linear $O(NW)$ performance, where the number of wavelets W is relatively insensitive to N.

Fig. 9 shows the expected inverse relation between variance and number of samples for the Monte Carlo method. The images in the top row show the power of final gather—Monte Carlo with 30 samples is noisy as expected, but is nearly smoothed out using one direct iteration. The bottom row shows that, as expected, pure Monte Carlo converges as the number of samples increases.

7 Experiments with Real Data

Our acquisition setup consists of a Dell 4310WX projector and a Canon EOS 5D Mark II camera. An accurate, one-time, radiometric calibration of the projector and camera is performed to ensure linearity of the corresponding signals [2]. We assemble 8 images at various exposures into a high dynamic range image.

We present two applications of our iterative light transport inversion — projector radiometric compensation and separating the bounces of light transport. As mentioned in Sec. 3, choosing $\mathbf{F}$ as the diagonal of $\mathbf{T}$ is accurate for radiometric compensation, even in non-Lambertian scenes. For our single projector-camera setup, that is only an approximation for applications like bounce

separation in specular scenes. However, higher bounces rapidly become diffuse in practice and our experiments show robust results even for non-diffuse scenes. We refer the reader to [2] for a more complete discussion.

Projector Radiometric Compensation: The ubiquitous use of projectors may necessitate inverting photometric distortions and interreflection effects to simulate any desired appearance in non-flat, non-Lambertian spaces. In terms of our theory, given a desired appearance $\mathbf{l}_{\text{out}}$, we seek to invert the light transport to find $\mathbf{l}_{\text{d}} = \mathbf{S}^{-1}\mathbf{l}_{\text{out}}$. As discussed in Sec. 3, we must account for the first bounce $\mathbf{F}$ from the projector, and actually compute $\mathbf{l}_{\text{in}} = \mathbf{T}^{-1}\mathbf{l}_{\text{out}}$.

Fig. 1 shows results for radiometric compensation to project a desired image onto a scene with non-Lambertian materials, occlusions and interreflections. Clearly, the desired appearance is closely matched. The size of the transport matrix is $131K \times 131K$, for which our iterative algorithm performs radiometric compensation in only about 3 secs. While such high resolutions may be infeasible for a straightforward matrix inversion, based on the patterns in Fig. 6, the stratified inverses method of [15] will require $1-2$ orders of magnitude more time. Also, in contrast to the method of [21], our algorithms are physically motivated and not contingent on any tunable parameters.

Separating Bounces: One consequence of our theory is that once the light transport has been acquired, we can quickly separate an image into the different bounces (direct, 1st bounce indirect, 2nd bounce indirect and so on). It follows from (19), noting that $\mathbf{S}^{-1} = \mathbf{I} - \mathbf{A}$, that the k-th indirect bounce is

$$\mathbf{l}_{\text{out}}^{(k+1)} - \mathbf{l}_{\text{out}}^{(k)} = \mathbf{l}_{\text{d}} - \mathbf{S}^{-1}\mathbf{l}_{\text{out}}^{(k)}. \tag{24}$$

Thus, each successive run of our iterative inversion algorithm yields a bounce of light transport. Fig. 10 shows a didactic example demonstrating the accuracy of the bounce separation. The scene consists of a white dihedral with green light projected on the left half. Note that successive bounces of indirect illumination in the bottom row alternate perfectly between the two walls, as expected. Fig. 11 demonstrates the same with a non-Lambertian occluder present in the scene. We observe that the specular highlight is limited only to the direct component and absent from the indirect bounces, which is also expected.

This application is the same as [18], but our algorithms are far more efficient. For instance, our iterative method recovers the direct component as well as each bounce of indirect illumination in 0.09 sec for the $4K \times 4K$ transport matrix in Fig. 11, while straightforward matrix inversion requires 4.6 sec. More importantly, our methods can efficiently operate on much higher resolution scenes that direct inversion cannot handle—for instance, Fig. 2 demonstrates bounce separation in a $131K \times 131K$ transport matrix. While an uncompressed matrix of that size cannot even be loaded in RAM, extrapolating from Fig. 6, a brute force inversion will require nearly 150 hours. In contrast, we require only 33ms per iteration in our (unoptimized) Matlab implementation, for a total of about 3 sec to separate each bounce. Note that the faster method of [13] yields only the top row of Fig. 10 for a *particular* lighting configuration, while we can separate all the bounces for *any* lighting, albeit at the expense of a more laborious acquisition.

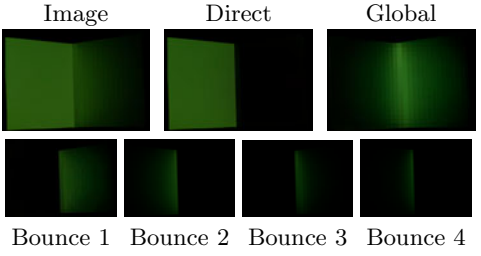


Fig. 10. Separation of individual bounces. The scene is a white concave dihedral, with flat green projection on the left half. **Top row:** input image and separated direct and net global components. **Bottom row:** recovered indirect bounces. Note that successive bounces illuminate alternating walls of the dihedral, as expected.

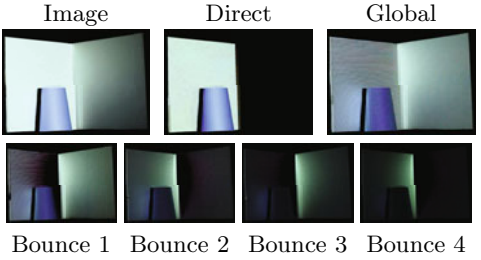


Fig. 11. Bounce separation with occlusions and specularities. **Top row:** input image and separated direct and net global components. **Bottom row:** recovered indirect bounces. Note that successive bounces illuminate alternating walls and the specular highlight is present only in the direct component.

We share some restrictions with other projector-camera systems, such as shutter speeds limited by projector refresh rates, color bleeding and non-linear color mixing ratios. For radiometric compensation, the projector cannot display negative values, which may lead to clipping artifacts in dark regions.

8 Conclusions and Future Work

The main contribution of this paper is a formulation of inverse light transport in computer vision, as a dual to the theory of forward rendering in computer graphics. This lends new insights for canceling interreflections in complex scenes, as well as fast computational methods for doing so. Our efficient algorithms, analogous to finite element radiosity and Monte Carlo path tracing in forward rendering, can handle transport resolutions far higher than previous methods.

From a theoretical perspective, we have just scratched the surface of analogies between forward and inverse methods. It is our hope that the framework of this paper forms the basis for discovering further insights into the structure of light transport and developing methods that couple fast acquisition and iterative inversion to perform radiometric compensation in dynamic scenes.

Acknowledgments. This work is funded by ONR YIP grant N00014-10-1-0032, ONR PECASE grant N00014-09-1-0741, a National Science Scholarship from A*STAR Graduate Academy of Singapore, as well as generous support from Adobe, NVIDIA, Intel and Pixar. We thank Joo Hwee Lim and Zhiyong Huang for kind support at I^2R and anonymous reviewers for useful comments.

References

1. Arvo, J., Torrance, K., Smits, B.: A framework for the analysis of error in global illumination algorithms. In: SIGGRAPH 1994, pp. 75–84 (1994)
2. Bai, J., Chandraker, M., Ng, T.-T., Ramamoorthi, R.: A dual theory of inverse and forward light transport. Technical Report UCB/EECS-2010-101, UC Berkeley (June 2010)
3. Bimber, O., Grundhoefer, A., Zeidler, T., Danch, D., Kapakos, P.: Compensating indirect scattering for immersive and semi-immersive projection displays. In: IEEE Virtual Reality, pp. 151–158 (2006)
4. Cohen, M., Wallace, J.: Radiosity and Realistic Image Synthesis. Academic Press, London (1993)
5. Ding, Y., Xiao, J., Tan, K.-H., Yu, J.: Catadioptric projectors. In: CVPR (2009)
6. Fujii, K., Grossberg, M., Nayar, S.: A Projector-camera System with Real-time Photometric Adaptation for Dynamic Environments. In: CVPR 2005, pp. 814–821 (2005)
7. Garg, G., Talvala, E., Levoy, M., Lensch, H.: Symmetric photography: Exploiting data-sparseness in reflectance fields. In: EGSR 2006, pp. 251–262 (2006)
8. Gortler, S., Schröder, P., Cohen, M., Hanrahan, P.: Wavelet radiosity. In: SIGGRAPH 1993, pp. 221–230 (1993)
9. Kajiya, J.: The rendering equation. In: SIGGRAPH 1986, pp. 143–150 (1986)
10. Marschner, S.: Inverse Rendering for Computer Graphics. PhD thesis, Cornell Univ. (1998)
11. Masselus, V., Peers, P., Dutre, P., Willems, Y.: Relighting with 4D incident light fields. ACM Trans. on Graphics (SIGGRAPH 2003) 22(3), 613–620 (2003)
12. Mukaigawa, Y., Kakinuma, T., Ohta, Y.: Analytical compensation of inter-reflection for pattern projection. In: ACM VRST, pp. 265–268 (2006)
13. Nayar, S., Krishnan, G., Grossberg, M., Raskar, R.: Fast separation of direct and global components of a scene using high frequency illumination. ACM Trans. on Graphics (SIGGRAPH 2006) 25(3), 935–944 (2006)
14. Nayar, S., Peri, H., Grossberg, M., Belhumeur, P.: A Projection System with Radiometric Compensation for Screen Imperfections. In: IEEE PROCAMS (2003)
15. Ng, T.-T., Pahwa, R.S., Bai, J., Quek, Q.-S., Tan, K.-H.: Radiometric Compensation Using Stratified Inverses. In: ICCV (2009)
16. Peers, P., Berge, K., Matusik, W., Ramamoorthi, R., Lawrence, J., Rusinkiewicz, S., Dutre, P.: A compact factored representation of heterogeneous subsurface scattering. ACM Trans. on Graphics (SIGGRAPH 2006) 25(3), 746–753 (2006)
17. Ramamoorthi, R., Hanrahan, P.: A signal-processing framework for inverse rendering. In: SIGGRAPH 2001, pp. 117–128 (2001)
18. Seitz, S., Matsushita, Y., Kutulakos, K.: A theory of inverse light transport. In: ICCV 2005, pp. 1440–1447 (2005)
19. Sen, P., Chen, B., Garg, G., Marschner, S., Horowitz, M., Levoy, M., Lensch, H.: Dual Photography. ACM Trans. on Graphics (SIGGRAPH) 24(3), 745–755 (2005)
20. Veach, E.: Robust Monte Carlo Methods for Light Transport Simulation. PhD thesis, Stanford University (1998)
21. Wetzstein, G., Bimber, O.: Radiometric Compensation through Inverse Light Transport. In: Pacific Conf. on Comp. Graphics and Appl., pp. 391–399 (2007)
22. Yu, Y., Debevec, P., Malik, J., Hawkins, T.: Inverse global illumination: Recovering reflectance models of real scenes from photographs. In: SIGGRAPH 1999, pp. 215–224 (1999)

Lighting Aware Preprocessing for Face Recognition across Varying Illumination

Hu Han[1,2], Shiguang Shan[1], Laiyun Qing[2], Xilin Chen[1], and Wen Gao[1,3]

[1] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China
[2] Graduate University of Chinese Academy of Sciences, Beijing 100049, China
[3] Institute of Digital Media, Peking University, Beijing 100871, China
{hhan,sgshan,lyqing,xlchen,wgao}@jdl.ac.cn

Abstract. Illumination variation is one of intractable yet crucial problems in face recognition and many lighting normalization approaches have been proposed in the past decades. Nevertheless, most of them preprocess all the face images in the same way thus without considering the specific lighting in each face image. In this paper, we propose a lighting aware preprocessing (LAP) method, which performs adaptive preprocessing for each testing image according to its lighting attribute. Specifically, the lighting attribute of a testing face image is first estimated by using spherical harmonic model. Then, a von Mises-Fisher (vMF) distribution learnt from a training set is exploited to model the probability that the estimated lighting belongs to normal lighting. Based on this probability, adaptive preprocessing is performed to normalize the lighting variation in the input image. Extensive experiments on Extended YaleB and Multi-PIE face databases show the effectiveness of our proposed method.

1 Introduction

Face recognition has attracted much attention in the past decades for its wide potential applications in commerce and law enforcement [1]. The challenges that a face recognition system has to face include variations in lighting, head pose, facial expression, accessory and so on. Among these factors, varying lighting conditions such as shadows, underexposure and overexposure in face imaging are intractable yet crucial problems that a practical face recognition system has to deal with. In the last decades, many approaches have been proposed to handle illumination variation problem with the goal of illumination normalization, illumination-insensitive feature extraction or illumination variation modeling. Among these approaches, many are based on image processing technique for the reason of simplicity and efficiency. In this paper, we refer these image processing based approaches as illumination preprocessing, and briefly review them in the following.

Histogram equalization (HE) [2] is one of the simplest illumination preprocessing approaches for face images, which can enhance the global contrast of one image. Logarithmic transformation (LT) [3], as a nonlinear transformation,

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 308–321, 2010.

tends to squeeze together the larger intensity values and stretch out the smaller ones in a face image. Jobson et al. [4] extended Retinex theory [5] to a single-scale Retinex (SSR) approach which could be used to enhance face images in improving local contrast and lightness. Based on the gamma correction technique that is widely used in Computer Graphics (CG), Shan et al. [6] proposed gamma intensity correction (GIC) in order to correct the overall brightness of a face image in accordance with a pre-defined face image with canonical lighting. Through analyzing the relationship between quotient image (QI) [7] algorithm and Retinex theory based on the reflectance-illumination model, Wang et al. [8] proposed self-quotient image (SQI) to handle the varying lighting conditions in face recognition without using a bootstrap set. Nishiyama and Yamaguchi [9] extended SQI as classified appearance-based quotient image (CAQI) in order to handle face regions with different albedo separately. Xie and Lam [10] proposed local normalization (LN) to reduce or remove the effect of uneven lighting conditions in order to get the corresponding face images under normal lighting. Considering that illumination variation mainly lies in the low-frequency band, Chen et al. [11] discarded an appropriate proportion of DCT coefficients in zigzag pattern in order to minimize the variation of face images from the same individual under different lighting conditions and then inverse DCT transform was performed to get the final illumination normalized images. Based on the reflectance-illumination imaging model, TV-L1 [12] model was introduced and analyzed in logarithm domain (LTV) by Chen et al. [13] for the purpose of decomposing a face image into large-scale and small-scale components, which correspond to illumination variation and intrinsic facial features respectively. And then only the small-scale features were used for face recognition. Xie et al. [14] reconstructed the illumination normalized face image by combining both the normalized large-scale component and smoothed small-scale component (RLS). Recently, face recognition using multi-band features are studied by Di et al. [15]. Tan and Triggs [16] presented a simple and efficient image preprocessing (PP) chain, which incorporated a series of steps such as gamma correction, Difference of Gaussian (DoG), masking and contrast equalization in order to extract illumination insensitive features for face recognition. However, most of the above approaches tend to perform illumination preprocessing equally on all the face images regardless of the particular lighting of each face image. This implies that a face image with canonical lighting will be processed like a face image with side lighting using completely the same parameter settings.

Intuitively, the above pattern that most of the existing lighting normalization approaches used to handle different lighting conditions is not optimal, since any preprocessing might bring negative effect if the input image is captured under normal lighting conditions. To reveal this possibility empirically, nine of the above-mentioned illumination preprocessing approaches, i.e., HE [2], LT [3], SSR [4], GIC [6], SQI [8], LN [10], DCT [11], LTV [13] and PP [16], are evaluated on Extended YaleB face database [17] in a traditional lighting unaware pattern. And Fisherfaces [18] is exploited as the recognition method following different illumination preprocessing approaches in our evaluation. The measurement of

the evaluation is the percentage of correcting originally-wrong matches (denoted as "positive") and reversing originally-correct matches (denoted as "negative"). The results are shown in Fig. 1, from which it is clear that most of the methods do bring some negative effects while improving the face recognition performance. Some of them may even completely counteract the positive effect, which thus limit the effectiveness of traditional lighting preprocessing approaches in improving variable lighting face recognition performance. Please note that, similar empirical observation was also reported in [19], which found that some of the preprocessing methods might result in lower recognition rates if applied to images with normal lighting.

Mathematically, most of the existing lighting normalization approaches try to have a universal method to deal with various cases. However, an image is a mapping of an object under certain lighting condition. To understand all these factors from a single image is an ill-posed problem. This is why most existing approaches reversed originally-correct matches in performing lighting normalization. In fact, it makes more sense to partition a problem as several sub-problems in handling an ill-posed problem.

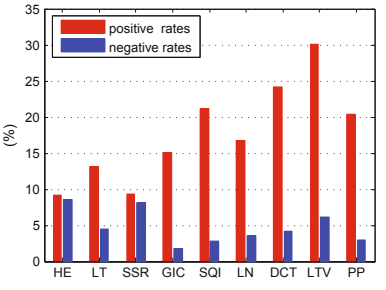


Fig. 1. The positive and negative effects of various illumination preprocessing methods performed in a lighting unaware way. We claim a "negative" if a face image is correctly recognized before the given preprocessing but incorrectly recognized after the preprocessing. On the contrary, a "positive" is reported if a face image originally incorrectly recognized can be correctly recognized after the specific preprocessing.

Based on the mathematical analysis above, we come to the idea that lighting normalization should be performed adaptively, and thus propose a lighting aware preprocessing (LAP) method for illumination-robust face recognition. Different from CAQI, in LAP, face images with different lighting conditions will be normalized in an adaptive preprocessing approach, i.e. face images with normal lighting will undergo minor or no illumination normalization, while face images with side lighting or abnormal exposure will be normalized by eliminating more large-scale components corresponding to lighting variations.

The remainder of this paper is structured as follows: Section 2 details the algorithm of the LAP and then extensive experiments are performed to verify the proposed approach in Sect. 3. Finally, we conclude this work in Sect. 4.

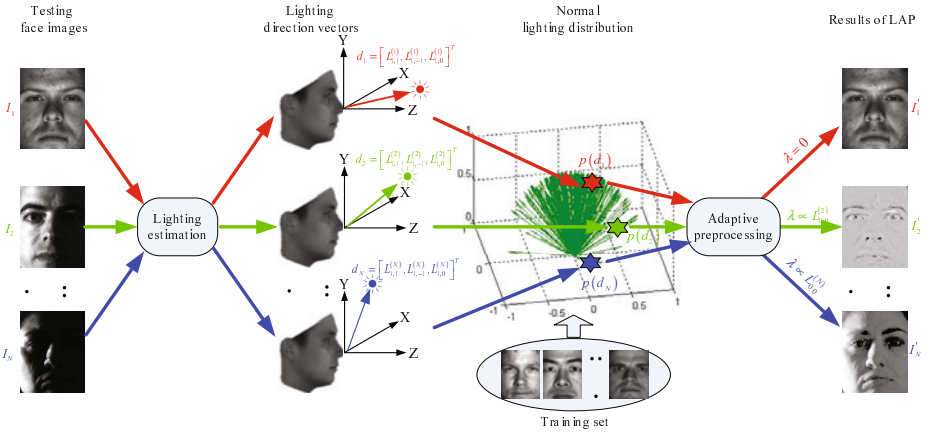


Fig. 2. Illustration for the framework of lighting aware preprocessing

2 Lighting Aware Preprocessing

In this section, we describe the details of the proposed LAP method. The algorithm overview of our LAP is shown in Fig. 2. Firstly, the lighting attribution of a testing face image is estimated by using spherical harmonic model. The estimated lighting is then analyzed by modeling the probability that it belongs to normal lighting. Finally, adaptive preprocessing is proposed to perform lighting normalization for the images with different lighting conditions. Details of each step are described below.

2.1 Lighting Attribute Estimation by Using Spherical Harmonic Model

As above mentioned, face images should be adaptively preprocessed according to their lighting conditions. Therefore, the lighting in each face image should be estimated. With different constraints introduced, many approaches have been proposed to recover the lighting from a single input face image, such as shape from shading (SFS) [20], 3D subspaces [21], 5D subspace [22], 9D linear subspace[23], illumination cone [17,24] and so on. In our LAP approach, lighting attribute is estimated by using spherical harmonic model, which has been used to estimate the harmonic basis face images that span a linear subspace to approximate a wide variety of illumination variations [23,25,26,27,28,29].

By simplifying the face imaging procedure as a convex Lambertian object under distant isotropic illumination, the image intensity is proportional to the radiance reflected by the face surface and can be approximated by

$$I(x, y) \approx \lambda(x, y) E(\alpha(x, y), \beta(x, y)) \tag{1}$$

where (x, y) ranges over the whole face surface, $\lambda(x, y)$ is the albedo at point (x, y), (α, β) is the normal at point (x, y) and $E(\alpha, \beta)$ is the total irradiance that arrives at point (x, y), which is a function of the surface normal (α, β) [30]

$$E(\alpha, \beta) = \int_{\varphi_i=0}^{2\pi} \int_{\theta_i=0}^{\pi/2} L_i((x, y), \varphi_i, \theta_i) \cos\theta_i \sin\theta_i d\theta_i d\varphi_i \tag{2}$$

where θ_i and φ_i are respectively the elevation and azimuth angles of incident light. Under the distant illumination assumption, $E(\alpha, \beta)$ is independent of surface position (x, y) [25]

$$E(\alpha, \beta) = \int_{\varphi_i=0}^{2\pi} \int_{\theta_i=0}^{\pi/2} L_i(\varphi_i, \theta_i) \cos\theta_i \sin\theta_i d\theta_i d\varphi_i \tag{3}$$

where $L_i(\varphi_i, \theta_i)$ is the radiance of the incident light with direction (φ_i, θ_i). Hence, lighting estimation is converted to recovering the coefficients $L_i(\varphi_i, \theta_i)$ given an input face image I.

As is shown independently by Basri and Jacobs [23] as well as Ramamoorthi and Hanrahan [25], $E(\alpha, \beta)$ can be well approximated by a combination of the first nine spherical harmonics

$$E(\alpha, \beta) = \sum_{l=0}^{2} \sum_{m=-l}^{l} \left(\frac{4\pi}{2l+1}\right)^{1/2} A_l L_{l,m} Y_{l,m}(\alpha, \beta) \tag{4}$$

where A_l is the spherical harmonic coefficient for transfer function, $L_{l,m}$ is the coefficient of incident lighting and $Y_{l,m}$ forms the orthonormal spherical harmonic basis. It is more convenient to parameterize $Y_{l,m}$ in Cartesian coordinate system as below [23]

$$\begin{aligned}
&Y_{0,0} = \sqrt{\tfrac{1}{4\pi}} && Y_{1,-1} = \sqrt{\tfrac{3}{4\pi}} y \\
&Y_{1,0} = \sqrt{\tfrac{3}{4\pi}} z && Y_{1,1} = \sqrt{\tfrac{3}{4\pi}} x \\
&Y_{2,-2} = \sqrt{\tfrac{15}{4\pi}} xy && Y_{2,-1} = \sqrt{\tfrac{15}{4\pi}} yz \\
&Y_{2,0} = \sqrt{\tfrac{5}{16\pi}} (3z^2 - 1) && Y_{2,1} = \sqrt{\tfrac{15}{4\pi}} zx \\
&Y_{2,2} = \sqrt{\tfrac{5}{16\pi}} (x^2 - y^2)
\end{aligned} \tag{5}$$

where (x, y, z) is the representation for surface normal (α, β) in Cartesian coordinate system. Combining (1) with (4), we will get

$$\begin{aligned}
I(x, y) &\approx \sum_{l=0}^{2} \sum_{m=-l}^{l} \left(\tfrac{4\pi}{2l+1}\right)^{1/2} \lambda(x, y) A_l L_{l,m} Y_{l,m}(\alpha(x, y), \beta(x, y)) \\
&= \sum_{l=0}^{2} \sum_{m=-l}^{l} L_{l,m} b_{l,m}(x, y)
\end{aligned} \tag{6}$$

where $b_{l,m}(x, y)$ is the harmonic image of a face

$$b_{l,m}(x, y) = \left(\frac{4\pi}{2l+1}\right)^{1/2} \lambda(x, y) A_l Y_{l,m}(\alpha(x, y), \beta(x, y)) \tag{7}$$

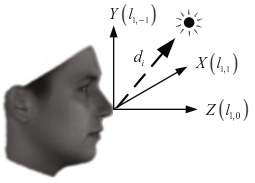


Fig. 3. The construction of a lighting direction vector

In order to estimate the 9 illumination coefficients $L_{l,m}$, one needs to know the albedo map $\lambda(x, y)$ and normal map $(\alpha(x, y), \beta(x, y))$ of the given face. However, in practice, they are usually unavailable for a single input face image. Fortunately, as shown in [27], with a quasi-constant albedo map and a warped generic 3D facial normal map as the approximations for the real ones, the 9 illumination coefficients $L_{l,m}$ can be well estimated by solving the following least squares problem

$$\hat{L} = \arg\min_{L} \|I - BL\|_{L_2} \tag{8}$$

where image I is vectorized as a P-dimensional column vector, B is a $P \times 9$ matrix with $b_{l,m}$ as its columns.

In our implementation, given an input face image, its two eyes are first localized and used to roughly align a generic 3D facial normal map. Then, spherical harmonic images, i.e. B, of this face are computed based on (7). And finally the 9 coefficients are estimated by solving (8).

According to the spherical harmonics theory, among the 9 illumination coefficients, $L_{0,0}$ is the DC component reflecting the average energy of the incident lighting, while the three first-order coefficients, $L_{1,1}, L_{1,-1}, L_{1,0}$, as illustrated in Fig. 3, reflect the intensity of incident lights in X, Y, Z directions respectively. Therefore, they are utilized in our method to form the lighting direction vector $d^{'} = [L_{1,1}, L_{1,-1}, L_{1,0}]^T$. Since we care only the relative quantity of these coefficients, we further normalize it by dividing its module and thus get a unit vector in L_2 norm $d = d^{'}/\|d^{'}\| = [l_{1,1}, l_{1,-1}, l_{1,0}]^T$, which is then used to analyze the lighting condition of the input image in the following.

2.2 Lighting Analysis with vMF Model

With the above estimated lighting attribute, what we need to do next is determining which kind of lighting it belongs to. However, it is difficult to make a quantitative definition of lighting category, as lighting condition is a subjective concept. To overcome the uncertainty of imaging procedure and the subjectiveness in lighting condition definition, we apply a statistical model to determine the probability that the estimated lighting belongs to normal lighting. The statistical model, which combines the principle of physics, geometrical model and the robustness of statistics, thus provides a relative definition of different lighting conditions instead of an absolute one.

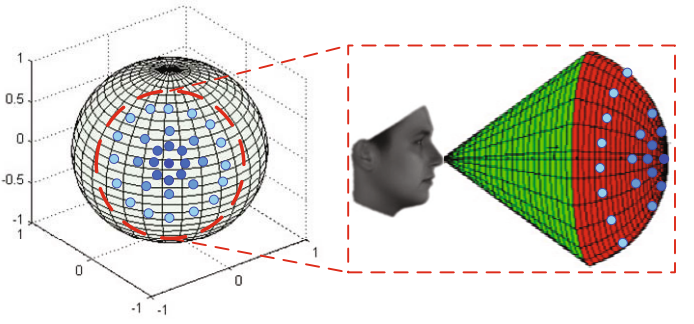


Fig. 4. The distribution of normal lighting is analogous to a Gaussian distribution on a sphere surface

In most face recognition testing protocols [17,31], so-called normal lighting usually means frontal distant lighting. Therefore, the subset with normal lighting in the testing protocol of each face database is utilized to learn a statistical model for normal lighting. In practice, normal lighting should distribute analogously to a Gaussian distribution on a unit sphere as illustrated in Fig. 4 with $d_0 = [0, 0, 1]^T$ being the expectation. Thus, normal lighting can be modeled as a von Mises-Fisher (vMF) distribution [32] which is widely used in directional statistics.

Specifically, a 3-dimensional unit random vector x(i.e., $x \in \mathbb{R}^3$ and $\|x\| = 1$) is of 3-variate von Mises-Fisher distribution if its probability density function is with the form

$$p(x|\mu, \kappa) = c(\kappa) \exp(\kappa \mu^T x) \tag{9}$$

where μ is the mean direction with $\|x\| = 1$, $\kappa(\kappa \geq 0)$ is the concentration parameter describing how strongly the unit random vectors sampled from the distribution are concentrated toward the mean direction, and normalization constant $c(\kappa)$ is defined as

$$c(\kappa) = \frac{\kappa}{4\pi \sinh \kappa} = \frac{\kappa}{2\pi(e^{\kappa} - e^{-\kappa})} \tag{10}$$

Given the vMF model, modeling normal lighting is then to estimate the parameters of the vMF model. In this study, maximum likelihood estimation is adopted to estimate μ and κ from a learning dataset. Formally, given a training set containing N face images captured under "normal" lighting conditions, we estimate the lighting direction vectors by the method in Section 2.1 for all the training images and obtain

$$D = \{d_i \in \mathbb{R}^3, 1 \leq i \leq N\} \tag{11}$$

By safely assuming d_i to be independent with each other, we have the following likelihood

$$p(D|\mu, \kappa) = \prod_{i=1}^{N} p(d_i|\mu, \kappa) \tag{12}$$

And then the log-likelihood will be

$$\ln p(D|\mu,\kappa) = N \ln c(\kappa) + \kappa \mu^T t \tag{13}$$

where $t = \sum_{i=1}^{N} d_i$. In order to get the maximum likelihood estimates for μ and κ, Lagrange multipliers is used to maximize the log-likelihood objective function

$$\Lambda(\mu,\kappa,d_i,\lambda) = N \ln c(\kappa) + \kappa \mu^T t + \lambda(1-\mu^T\mu) \tag{14}$$

subject to the constraint $\mu^T\mu = 1(\|\mu\| = 1)$. Let the derivative $d\Lambda = 0$, then we get the following system of equations

$$\begin{array}{l} \frac{\partial \Lambda}{\partial \mu} = \kappa t - 2\lambda\mu \quad\quad = 0 \\ \frac{\partial \Lambda}{\partial \kappa} = \frac{N c^{'}(\kappa)}{c(\kappa)} + \mu^T t = 0 \\ \frac{\partial \Lambda}{\partial \lambda} = 1 - \mu^T\mu \quad\quad = 0 \end{array} \tag{15}$$

From (15), it is not difficult to get an estimate for μ

$$\hat{\mu} = \frac{t}{\|t\|} \tag{16}$$

In directional statistics, the concentration parameter κ is usually estimated in an approximation manner [32,33] and for a 3-variate von Mises-Fisher distribution, the following approximation will be sufficient

$$\hat{\kappa} = \frac{3\bar{t} - \bar{t}^3}{1-\bar{t}^2} \tag{17}$$

where $\bar{t} = \|t\|/N$

After μ and κ are estimated, the statistical model for describing normal lighting is constructed and then the probability that the estimated lighting d of a testing face image belongs to normal lighting can be calculated based on (9)

$$p(d|\hat{\mu},\hat{\kappa}) = c(\hat{\kappa}) \exp(\hat{\kappa}\hat{\mu}^T x) \tag{18}$$

In this paper, the subset#1 from Extended YaleB face database is used as the training set. The face images in subset#1 are captured with the angle between the light source direction and the camera axis within 12°. Details about the subset division for Extended YaleB can be found in [17].

2.3 Adaptive Lighting Preprocessing

As we have mentioned before, once the lighting condition of a testing face image has been grouped in a relative manner, facial images will be handled accordingly. For this purpose, we further propose an adaptive method to perform illumination normalization for each testing face image. By varying the truncation scale, many existing approaches, e.g. the Gaussian smoothing filter used in [4,8], the DCT

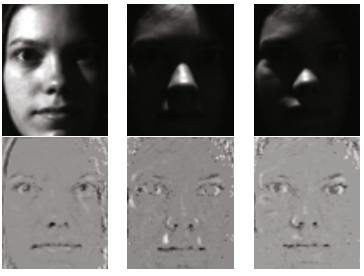


Fig. 5. The results of traditional LTV on three face images of one individual

reported in [11] and the TV-L1 model [12] utilized in LTV [13], can reach a better balance between eliminating extrinsic lighting variation and preserving intrinsic facial features. Without loss of generality, here TV-L1 model is used to implement adaptive preprocessing based on the estimated probability that the lighting in a testing face image belongs to normal lighting.

TV-L1 model aims at decomposing a face image into large-scale component u which corresponds to illumination variation and small-scale component v which corresponds to intrinsic facial feature and the large-scale component in a face image is estimated by solving the following variational problem

$$\hat{\mu} = \arg\min_{u} \int |\nabla u| + \lambda \|I - u\|_{L_1} \tag{19}$$

where $\int |\nabla u|$ is the total variation of u and λ is a scalar constant controlling the scale truncation. With u solved, the small-scale component v can be calculated as $v = I - u$, which can then be used for face recognition across varying lighting conditions. Evidently, in TV-L1 model, the scale-truncation constant λ actually balances the illumination removal by u and feature preserving in v. However, in LTV, it is empirically set and kept the same for all face images. This might be questionable, since different lighting attributes imply illumination component of different scales. Figure 5 shows some examples of LTV with fixed λ for images of the same person but with different lighting attributes. It is clear that the results are not desirable. Different from LTV, in our adaptive lighting preprocessing, TV-L1 model are applied in an adaptive pattern based on the above estimated probability rather than in a fixed pattern.

According to the analysis for parameter λ in [12], a larger truncation scale is more desirable in order to avoid discarding too much intrinsic facial features for face images with normal lighting and correspondingly a smaller λ should be used for TV-L1 model. While the effect introduced by abnormal lighting, such as the artificial edges caused by side lighting, mainly lies in high frequency band; therefore a small truncation scale is suitable and correspondingly a larger λ should be taken.

According to the above analysis, the parameter λ in TV-L1 model can be approximately determined based on the probability that the lighting in a face image belongs to normal lighting

$$\lambda = (1 - p(d|\hat{\mu}, \hat{\kappa}))\beta \tag{20}$$

where $p(d|\hat{\mu}, \hat{\kappa})$ is the above estimated probability that the lighting in a testing face image belongs to normal lighting, β is the range for parameter λ. In TV-L1 model, parameter λ can be set as any positive real number, but in practice, for face image with the size of 64×80, λ in the range of $[0, 1.2]$ will be sufficient for handling most of the lighting variations. A linear relationship between p and λ seems simple but reveals to be effective in our experiments. To be note that in the theory of TV-L1, features of all scales should be keep in v when $\lambda = 0$, i.e. $v_{\lambda=0} = I$; however, due to the limitation in computation, v cannot be calculated when $\lambda = 0$. Therefore, we force $v_{\lambda=0} = I$ in our implementation. When TV-L1 model is substituted by other methods, e.g. Gaussian smoothing filter or DCT, the parameters can also be determined like in (20). All the gallery images are also preprocessed in the same way as each testing face image when performing face recognition.

3 Experimental Results

3.1 Databases and Settings for Experiments

Extended YaleB [17], PIE [34] and Multi-PIE [31] are three representative face databases in the area, however, many illumination preprocessing approaches, including the proposed LAP, have gotten 100% recognition performance. Therefore, two challenging face databases of the three: Extended YaleB [17] and Multi-PIE [34], are exploited in our experiments to compare our proposed approach with other illumination preprocessing approaches in face recognition across varying illumination.

Extended YaleB face database includes the original YaleB face database with 10 individuals under 64 different illumination conditions and the extended part with 28 individuals that are also captured under 64 different illumination conditions. Totally 2,432 face images of 38 individuals under 64 illumination conditions in frontal view are used for experiments. All the face images are divided into five subsets according to [17], in which subset#1 is used as the training set for both lighting estimation and face recognition algorithm. The varying lighting in Extended YaleB is harsh for illumination-robust recognition as the lighting directions vary from left 130° degrees to right 130°.

Multi-PIE is a recently published face database, which contains as many as 755,370 images from 337 subjects, imaged under 15 view points and 19 illumination conditions in up to four recording sessions [31]. According to the testing protocol in [31], the face images of 14 randomly selected subjects are used for training and images of all the other 323 subjects are used for testing. Among all the testing images, only one face image of each individual recorded without flashes is used as gallery. The huge database size and time span of Multi-PIE have determined the challenge for variable lighting face recognition. Moreover, the limitation of 14 subjects for training further increase the difficulty in recognition across varying lighting conditions.

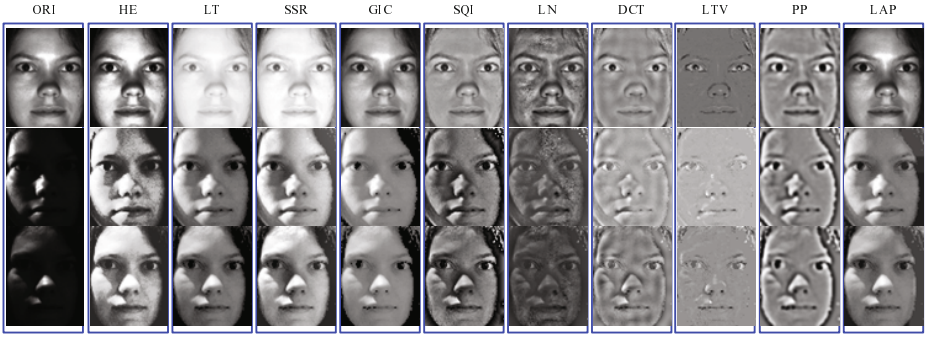


Fig. 6. Illumination preprocessing on testing face images using different approaches. Images in the first column are the original input face images under different lighting conditions. Images in the rest columns are the results of different illumination preprocessing approaches.

Before performing any illumination preprocessing, all the face images are geometrically normalized to the size of 64×80 with the distance between two eyes 35 pixels. The proposed LAP is a kind of illumination normalization approach instead of illumination-insensitive feature extraction or illumination variation modeling approach, therefore, the state of the art as well as several representative illumination normalization approaches are taken for comparison, i.e., HE [2], LT [3], SSR [4], GIC [6], SQI [8], LN [10], DCT [11], LTV [13] and PP [16]. For fair comparison with other methods, we exploited the parameters settings recommended in the original literature proposing the corresponding methods for comparison. As what we concern is the comparison between different lighting preprocessing approaches, Fisherfaces [18] is fixed as the recognition algorithm for all the illumination preprocessing approaches we compared.

Face recognition is performed on the illumination normalized face images preprocessed by different approaches and recognition performance is reported to verify the effectiveness of different lighting preprocessing approaches in improving the robustness for face recognition across varying lighting conditions. For the convenience of our description, we denote "ORI" as the original face images without any lighting preprocessing.

3.2 Comparisons

The visualization of some illumination normalized face image of different lighting preprocessing approaches is illustrated in Fig. 6. In the figure, the face images in the first column are the original input images and those in the rest columns are the results of different lighting preprocessing approaches labeled above the column. As can be seen from the figure, the traditional approaches performed in a lighting unaware pattern tend to produce satisfying results for some kinds of lighting but not for others. On the contrary, for our LAP, testing images with normal lighting are kept as close to the original as possible and images

Table 1. Face recognition performance of Fisherfaces on the preprocessed face images by different illumination preprocessing approaches from Extended YaleB and Multi-PIE databases

Approach	Recognition Rate (%)	
	Extended YaleB	Multi-PIE
ORI	54.15	52.77
HE	54.75	62.53
LT	62.79	62.73
SSR	55.45	63.79
GIC	67.73	64.27
SQI	72.58	65.71
LN	67.36	60.74
LDCT	74.10	61.82
LTV	78.02	60.81
PP	71.56	61.18
LAP	**86.89**	**71.15**

with abnormal exposure are processed to discard most lighting variations while preserving more discriminative facial features compared with LTV.

Face recognition experiments are then performed on the two face databases following different illumination preprocessing approaches and the recognition rates are reported in Table 1. As shown in the table, our LAP achieves impressively better face recognition performance than all the other methods on both Extended YaleB and Multi-PIE face databases. On Extended YaleB, compared with LTV, LAP gets more than 8% higher face recognition rate. Even on the much more challenging Multi-PIE face database, LAP gets the highest face recognition rate 71.15%. Experimental results on Extended YaleB and Multi-PIE face databases suggest that our proposed LAP framework is more effective and robust in improving face recognition performance across varying illumination compared with the traditional lighting unaware approaches.

4 Conclusions

Traditional illumination preprocessing methods deal with face images in a lighting unaware way, so they might suffer from negative effect, for instance, failing to recognize an image which can correctly recognized before preprocessing. This paper analyzed the problem and proposes a lighting aware preprocessing method. In the method, face images with different lighting conditions are processed according to the lighting attribute in the images. Experiments illustrate impressive performance improvement compared with the state of the art and representative illumination preprocessing methods. To be note that although TV-L1 is utilized in the proposed LAP framework, other methods such as low-pass filtering and DCT can also be embedded into the proposed LAP framework.

The preliminary studies in this paper show that we still have large space for improvement for illumination-invariant face recognition. Preprocessing automatically adapted to the lighting attribute of the image might be a promising possibility.

Currently, spherical harmonic model is used to estimate the lighting in a testing face image. Simple and efficient approaches without using 3D face information, e.g. the method proposed by S. Choi, et al. [35], might be used for lighting estimation. Moreover, the relationship between the normal lighting probability and adaptive parameter selection will also be exploited in future work.

Acknowledgments

This paper is partially supported by Natural Science Foundation of China under contracts No.60803084, No.60872077, and No. U0835005; National Basic Research Program of China (973 Program) under contract 2009CB320902, and ISVISION Technology Co. Ltd.

References

1. Zhao, W., Chellappa, R., Rosenfeld, A., Phillips, P.J.: Face recognition: A literature survey. ACM Computing Surveys 35, 399–458 (2003)
2. Gonzalez, R., Woods, R.: Digital image processing, pp. 91–94. Prentice Hall, USA (1992)
3. Adini, Y., Moses, Y., Ullman, S.: Face recognition: The problem of compensating for changes in illumination direction. IEEE Trans. PAMI 19, 721–732 (1997)
4. Jobson, D.J., Rahman, Z., Woodell, G.A.: Properties and performance of a center/surround retinex. IEEE Trans. IP 6, 451–462 (1997)
5. Land, E.H.: An alternative technique for the computation of the designator in the retinex theory of color vision. Proc. Nati. Acad. Sci. USA 83, 3078–3080 (1986)
6. Shan, S., Gao, W., Cao, B., Zhao, D.: Illumination normalization for robust face recognition against varying lighting conditions. In: Proc. AMFG, Nice, pp. 157–164 (2003)
7. Shashua, A., Raviv, T.R.: The quotient image: Class-based re-rendering and recognition with varying illuminations. IEEE Trans. PAMI 23, 129–139 (2001)
8. Wang, H., Li, S., Wang, Y.: Face recognition under varying lighting conditions using self quotient image. In: Proc. FG, Seoul, pp. 819–824 (2004)
9. Nishiyama, M., Yamaguchi, O.: Face recognition using the classified appearanceebased quotient image. In: Proc. FG, Southampton, pp. 49–54 (2006)
10. Xie, X., Lam, K.: An efficient illumination normalization method for face recognition. Pattern Recognition Letters 27, 609–617 (2006)
11. Chen, W., Er, M.J., Wu, S.: Illumination compensation and normalization for robust face recognition using discrete cosine transform in logarithm domain. IEEE Trans. SMC:B 36, 458–466 (2006)
12. Chan, T., Esedoglu, S.: Aspects of total variation regularized l1 function approximation. CAM Report, 4–7 (2004)
13. Chen, T., Yin, W., Zhou, X.S., Comaniciu, D., Huang, T.S.: Total variation models for variable lighting face recognition. IEEE Trans. PAMI 28, 1519–1524 (2006)
14. Xie, X., Zheng, W., Lai, J., Yuen, P.C.: Face illumination normalization on large and small scale features. In: Proc. CVPR, Alaska, pp. 1–8 (2008)
15. Di, W., Zhang, L., Zhang, D., Pan, Q.: Studies on hyperspectral face recognition with feature band selection. IEEE Trans. SMC-A (to appear)

16. Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. In: Proc. ICCV Workshop, Rio de Janeiro, pp. 168–182 (2007)
17. Georghiades, A.S., Belhumeur, P.N., Kriegman, D.J.: From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Trans. PAMI 23, 643–660 (2001)
18. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. IEEE Trans. PAMI 19, 711–720 (1997)
19. Du, B., Shan, S., Qing, L., Gao, W.: Empirical comparisons of several preprocessing methods for illumination insensitive face recognition. In: Proc. ICASSP, Pennsylvania, pp. 981–984 (2005)
20. Horn, B.K.P., Brooks, M.J.: The variational approach to shape from shading. CVGIP 33, 174–208 (1986)
21. Shashua, A.: On photometric issues in 3d visual recognition from a single 2d image. IJCV 21, 99–122 (1997)
22. Hallinan, P.W.: A low-dimensional representation of human faces for arbitrary lighting conditions. In: Proc. CVPR, Seattle, pp. 995–999 (1994)
23. Basri, R., Jacobs, D.W.: Lambertian reectance and linear subspaces. IEEE Trans. PAMI 25, 218–233 (2003)
24. Belhumeur, P.N., Kriegman, D.J.: What is the set of images of an object under all possible illumination conditions? IJCV 28, 245–260 (1998)
25. Ramamoorthi, R., Hanrahan, P.: On the relationship between radiance and irradiance: determining the illumination from images of a convex lambertian object. JOSA 18, 2448–2459 (2001)
26. Zhang, L., Samaras, D.: Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics. IEEE Trans. PAMI 28, 351–363 (2006)
27. Qing, L., Shan, S., Gao, W., Du, B.: Face recognition under generic illumination based on harmonic relighting. IJPRAI 19, 513–531 (2005)
28. Wang, Y., Liu, Z., Hua, G., Wen, Z., Zhang, Z., Samaras, D.: Face re-lighting from a single image under harsh lighting conditions. In: Proc. CVPR, Minnesota, pp. 1–8 (2007)
29. Jiang, X., Kong, Y.O., Huang, J., Zhao, R.-c., Zhang, Y.: Learning from real images to model lighting variations for face images. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 284–297. Springer, Heidelberg (2008)
30. Forsyth, D.A., Ponce, J.: Computer vision: A modern approach, pp. 46–58. Prentice Hall, USA (2002)
31. Gross, R., Matthews, I., Cohn, J., Kanade, T., Baker, S.: Multi-pie. Image and Vision Computing 28, 807–813 (2010)
32. Mardia, K.V., Jupp, P.E.: Directional statistics, pp. 36–44. J. Wiley, Chichester (2000)
33. Banerjee, A., Dhillon, I.S., Ghosh, J., Sra, S.: Clustering on the unit hypersphere using von mises-fisher distributions. JMLR 9, 1345–1382 (2005)
34. Sim, T., Baker, S., Bsat, M.: The cmu pose, illumination, and expression database. IEEE Trans. PAMI 25, 1615–1618 (2003)
35. Choi, S., Kim, C., Choi, C.: Shadow compensation in 2d images for face recognition. Pattern Recognition 40, 2118–2125 (2007)

Detecting Ground Shadows in Outdoor Consumer Photographs

Jean-François Lalonde, Alexei A. Efros, and Srinivasa G. Narasimhan

School of Computer Science, Carnegie Mellon University
http://graphics.cs.cmu.edu/projects/shadows

Abstract. Detecting shadows from images can significantly improve the performance of several vision tasks such as object detection and tracking. Recent approaches have mainly used illumination invariants which can fail severely when the qualities of the images are not very good, as is the case for most consumer-grade photographs, like those on Google or Flickr. We present a practical algorithm to automatically detect shadows cast by objects onto the ground, from a single consumer photograph. Our key hypothesis is that the types of materials constituting the ground in outdoor scenes is relatively limited, most commonly including asphalt, brick, stone, mud, grass, concrete, etc. As a result, the appearances of shadows on the ground are not as widely varying as general shadows and thus, can be learned from a labelled set of images. Our detector consists of a three-tier process including (a) training a decision tree classifier on a set of shadow sensitive features computed around each image edge, (b) a CRF-based optimization to group detected shadow edges to generate coherent shadow contours, and (c) incorporating any existing classifier that is specifically trained to detect grounds in images. Our results demonstrate good detection accuracy (85%) on several challenging images. Since most objects of interest to vision applications (like pedestrians, vehicles, signs) are attached to the ground, we believe that our detector can find wide applicability.

1 Introduction

Shadows are everywhere! Yet, the human visual system is so adept at filtering them out, that we never give shadows a second thought; that is until we need to deal with them in our algorithms. Since the very beginning of computer vision, the presence of shadows has been responsible for wreaking havoc on a wide variety of applications, including segmentation, object detection, scene analysis, stereo, tracking, etc. On the other hand, shadows play a crucial role in determining the type of illumination in the scene [1,2] and the shapes of objects that cast them [3]. But while standard approaches, software, and evaluation datasets exist for a wide range of important vision tasks, from edge detection to face recognition, there has been comparatively little work on shadows in the last 40 years. Approaches that use multiple images [4], time-lapse image sequences [5,6] or user inputs [7,8,9] have demonstrated impressive results, but detecting shadows

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 322–335, 2010.

reliably and automatically from a single image remains an open problem. This is because the appearances and shapes of shadows outdoors depend on several hidden factors such as the color, direction and size of the illuminants (sun, sky, clouds), the geometry of the objects that are casting the shadows and the shape and material properties of objects onto which the shadows are cast.

Most works for detecting shadows from a single image are based on computing illumination invariants that are physically-based and are functions of individual pixel values [10,11,12,13,14] or the values in a local image neighborhood [15]. Unfortunately, reliable computations of these invariants require high quality images with wide dynamic range, high intensity resolution and where the camera radiometry and color transformations are accurately measured and compensated for. Even slight perturbations (imperfections) in such images can cause the invariants to fail severely (see Fig. 4). Thus, they are ill-suited for the regular consumer-grade photographs such as those from Flickr and Google, that are noisy and often contain compression, resizing and aliasing artifacts, and effects due to automatic gain control and color balancing. Since much of current computer vision research is done on consumer photographs (and even worse-quality photos from the mobile phones), there is an acute need for a shadow detector that could work on such images.

Our goal is to build a reliable shadow detector for consumer photographs of outdoor scenes. While detecting all shadows is expected to remain hard, we explicitly focus on the shadows cast by objects onto the ground plane. Fortunately, the types of materials constituting the ground in typical outdoor scenes are (relatively) limited, most commonly including concrete, asphalt, grass, mud, stone, brick, etc. Given this observation, our key hypothesis is that the appearances of shadows on the ground are not as widely varying as the shadows everywhere in the scene and can be learned from a set of labelled images of real world scenes. This restriction by no means makes the problem trivial: the ground shadow detector still needs to contend with myriad other non-shadow visual manifestations such as markings and potholes on the roads, pavement/road boundaries, grass patterns on lawns, etc. Further, since many objects (pedestrians, vehicles, traffic signs, etc) of interest to vision applications, are attached to the ground and cast shadows onto the ground, we believe such a ground shadow detector will find wide applicability.

1.1 Overview

Our approach consists of three stages depending on the information in the image used. In the first stage, we will exploit local information around edges in the image. For this, we compute a set of shadow sensitive features that include the ratios of brightness and color filter responses at different scales and orientations on both sides of the edge. These features are then used with a trained decision tree classifier to detect whether an edge is a shadow or not. The idea is that while any single feature may not be useful for detecting all ground shadows, the classifier is powerful enough to choose the right features depending on the underlying edge region. In order to make the classifier robust to non-shadow

edges, a negative training set is constructed from a set of edges not on the ground and those arising due to road markings, potholes, grass/mud boundaries, etc. Surprisingly, this simple procedure yields 80% classification accuracy on our test set of images randomly chosen from Flickr and LabelMe [16].

In the second stage, we enforce a grouping of the shadow edges using a Conditional Random Field (CRF) to create longer contours. This is similar in spirit to the classical constrained label propagation used in mid-level vision tasks [17]. This procedure connects likely shadow edges, discourages T-junctions which are highly unlikely on shadow boundaries, and removes isolated weak edges. But how do we detect the ground in an image? For this, in the third stage, we incorporate a global scene layout descriptor within our CRF, such as the 3-way ground-vertical surface-sky classifier by Hoiem et. al [18]. Since the scene layout classifier is trained on the general features of the scene and not the shadows, we are able to reduce the number of false-positive (non-shadow) detections outside the ground. Our results show that the shadow detection results improve by 5% with this step.

We demonstrate successful shadow detection on several images of natural scenes that include beaches, meadows and forest trails, as well as urban scenes that include numerous pedestrians, vehicles, trees, roads and buildings, captured under a variety of illumination conditions (sunny, partly cloudy, overcast). Similarly to the approach of Zhu et al. [19], our method relies on learning the appearance of shadows based on image features, but does so by using full color information. We found that using color features and incorporating knowledge of the ground location improve classification results as much as 10% on our test set. While our technique can be used as a stand-alone shadow detector, we believe it can also be tightly integrated into higher level scene understanding tasks.

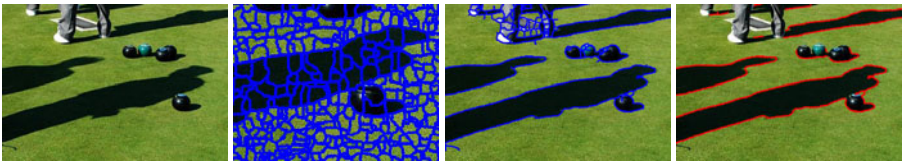
2 Learning Local Cues for Shadow Detection

Our approach relies on a classifier which is trained to recognize ground shadow edges by using features computed over a local neighborhood around the edge. We show that it is indeed possible to obtain good classification accuracy by relying on local cues, and that it can be used as a building block for subsequent steps. In this section, we describe how to build, train, and evaluate such a classifier.

2.1 From Pixels to Boundaries

We first describe the underlying representation on which we compute features. Since working with individual pixels is prone to noise and computationally expensive, we propose to instead reason about *boundaries*, or groups of pixels along an edge in the image. To obtain these boundaries, we first smooth the image with a bilateral filter [20], compute gradient magnitudes on the filtered image, and then apply the watershed segmentation algorithm on the gradient map. Fig. 1(b) shows a close-up example of such boundaries.

An undesirable consequence of the watershed segmentation is that it generates boundaries in smooth regions of the image (Fig. 1(b)). To compensate for this,



(a) Input image (b) Boundaries (c) Strong boundaries (d) Output

Fig. 1. Processing stages for the local classifier. The input image (a) is over-segmented into thousands of regions to obtain boundaries (b). Weak boundaries are filtered out by a Canny edge detector (c), and the classifier is applied on the remainder. (d) shows the boundaries i for which $P(y_i = 1|\mathbf{x}) > 0.5$. Note the correct classification of occlusion contours around the person's legs and the reflectance edges in the white square between the person's feet.

we retain only those boundaries which align with the strong edges in the image. For this, we use the canny edge detector at 4 scales to account for blurry shadow edges ($\sigma^2 = \{1, 2, 4, 8\}$), with a high threshold empirically set to $t = 0.3$. Under these conditions, we verified that the initial set of boundaries contain more than 97% of the true shadow edges in our dataset. For example, Fig. 1(c) shows the set of boundaries on which our classifier is evaluated for that image.

2.2 Local Shadow Features

We now describe the features computed over each boundary in the image. A useful feature to describe a shadow edge is the ratio of color intensities on both sides of the edge (e.g. min divided by max) [21]. The intuition is that shadows should have a specific ratio that is more or less the same across an image, since it is primarily due to the differences in natural lighting inside and outside the shadow. Since it is hard to manually determine the best color space [22] or best scale to compute features, we use 3 different colors spaces (RGB, LAB, [23]) and 4 different scales, and let the classifier automatically select the relevant features during the training phase. Although color-based features are bound to be affected by camera non-linearities, we found these ratios to work well across a wide range of cameras and capture conditions.

For a pixel along a boundary, we compute the intensity on one side of the edge (say, the left) by evaluating a weighted sum of pixels on the left of the edge. But which pixels to choose? We could use the watershed segments, but they do not typically extend very far. Instead, we use an oriented gaussian derivative filter of variance σ^2, but keep only its values which are greater than zero. We align the filter with the boundary orientation such that its positive weights lie on the left of the boundary and convolve it with the image to obtain f_l. The same operation is repeated with the filter rotated by 180° to obtain the weighted mean of pixels on its right f_r. Color ratios can then be computed at pixel p by $\frac{\min(f_l(p), f_r(p))}{\max(f_l(p), f_r(p))}$. This is done independently for each color channel of the RGB, LAB, and illumination-invariant [23] color spaces. To account for edge sharpness, we compute each filter at 4 different scales $\sigma^2 = \{1, 2, 4, 8\}$ and size $2\sigma^2$, to obtain 36 ratios in total.

We also employ two features suggested in [19] which capture the texture and intensity distribution differences on both sides of a boundary. The first feature computes a histogram of textons at 4 different scales, and compares them using the χ^2-distance. The texton dictionary was computed on a non-overlapping set of images. The second feature computes the difference in skewness of pixel intensities, again at the same 4 scales.

Finally, we concatenate the absolute value of the minimum filter response $\min(f_l(p), f_r(p))$ computed over the intensity channel to obtain the final, over-complete, 48-dimensional feature vector at every pixel. Boundary feature vectors are obtained by averaging the features of all pixels that belong to it.

2.3 Classifier

Having computed the feature vector $\mathbf{x}_i$ at each strong boundary in the image, we can now use them to train a classifier to learn the probability $P(y_i|\mathbf{x}_i)$ that boundary i is due to a shadow (which we denote with label y_i). We estimate that distribution using a logistic regression version of Adaboost [24], with twenty 16-node decision trees as weak learners. This classification method provides good feature selection and outputs probabilities, and has been successfully used in a variety of other vision tasks [18,25].

To train the classifier, we selected 170 images from LabelMe [16], Flickr, and the dataset introduced in [19], with the only conditions being that the ground must be visible, and there must be shadows. The positive training set contains manually labelled shadow boundaries, while the negative training set is populated with an equal amount of strong non-shadow boundaries on the ground (e.g. street markings) and occlusion boundaries.

We obtain a per-boundary classification accuracy of 79.7% (chance is 50%, see Fig. 5 for a breakdown per class). See Fig. 1(d) for an example. This result support out hypothesis: while the appearance of shadows on *any* type of material in *any* condition might be impossible to learn, the space of shadow appearances on the ground in outdoor scenes may not be that large after all!

3 Creating Shadow Contours

Despite encouraging results, our classifier is limited by its locality since it treats each boundary independently of the next. However, the color ratios of a shadow boundary should be consistent with those of its neighbors, since the sources illuminating nearby scene points should also be similar. Thus, we can exploit higher order dependencies across local boundaries to create longer shadow contours as well as remove isolated/spurious ones.

To model these dependencies, we construct a graph with individual boundaries as nodes (such as those in Fig. 1(b)) and drawing an edge across boundaries which meet at a junction point. We then define a CRF on that graph, which expresses the log-likelihood of a particular labeling $\mathbf{y}$ (i.e. assignment of

(a) (b)

Fig. 2. Creating shadow contours by enforcing local consistency. Our CRF formulation may help to (a) bridge the gap across X-junctions where the local shadow classifier might be uncertain, and (b) remove spurious T-junctions which should not be caused by shadows.

shadow/non-shadow to each boundary) given observed data $\mathbf{x}$ as a sum of unary $\phi_i(y_i)$ and pairwise potentials $\psi_{i,j}(y_i, y_j)$:

$$-\log P(\mathbf{y}|\mathbf{x};\lambda,\beta) = \lambda \sum_{i\in\mathcal{B}} \phi_i(y_i) + \sum_{(i,j)\in\mathcal{E}} \psi_{i,j}(y_i,y_j) - \log Z_{\lambda,\beta} \ , \qquad (1)$$

where $\mathcal{B}$ is the set of boundaries, $\mathcal{E}$ the set of edges between them, and λ and β are model parameters. In particular, λ is a weight controlling the relative importance of the two terms. $Z_{\lambda,\beta}$ is the partition function that depends on the parameters λ and β, but not on the labeling $\mathbf{y}$ itself.

Intuitively, we would like the unary potentials to penalize the assignment of the "shadow" label to boundaries which are not likely to be shadows according to our local classifier. This can be modeled using

$$\phi_i(y_i) = -\log P(y_i|\mathbf{x}_i) \ . \qquad (2)$$

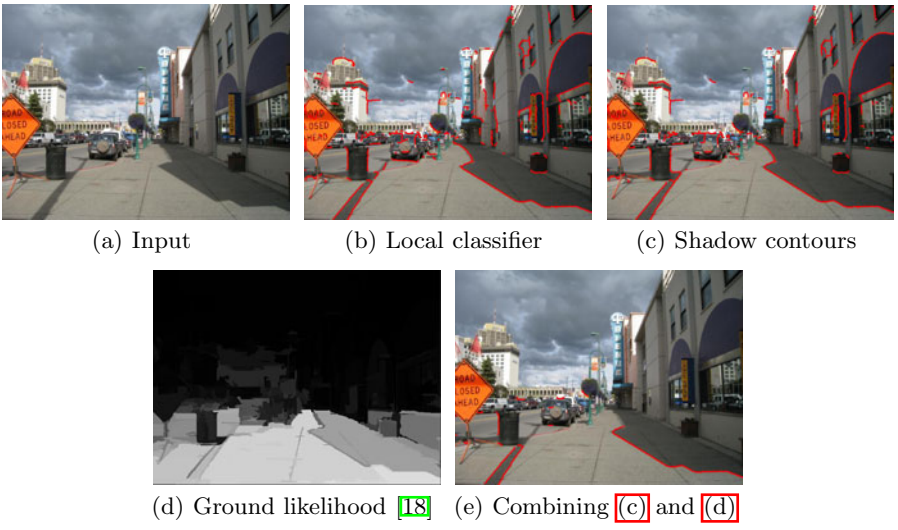
We would also like the pairwise potentials to penalize the assignment of different labels to neighboring boundaries that have similar features, which can be written as

$$\psi_{i,j}(y_i,y_j) = \mathbf{1}(y_i \neq y_j) \exp(-\beta \|\mathbf{x}_i - \mathbf{x}_j\|_2^2) \ , \qquad (3)$$

where $\mathbf{1}(\cdot)$ is the indicator function, and β is a contrast-normalization constant as suggested in [26]. In other words, we encourage neighboring shadows which have similar features and strong local probabilities to be labelled as shadows.

The negative likelihood in (1) can be efficiently minimized using graph cuts [27,28,29]. The free parameters were assigned the values of $\lambda = 0.5$ and $\beta = 16$ obtained by 2-fold cross-validation on a non-overlapping set of images.

Applying the CRF on our test images results in an improvement of roughly 1% in total classification accuracy, for a combined score of 80.5% (see Fig. 5-(b)). But more importantly, in practice, the way the CRF is setup encourages continuity, crossing through X-junctions, and discourages T-junctions as shown in Fig. 2. Since shadows are usually signaled by the presence of X-junctions and the absence of T-junctions [30], this reduces the number of false positives.



(a) Input (b) Local classifier (c) Shadow contours

(d) Ground likelihood [18] (e) Combining (c) and (d)

Fig. 3. Incorporating scene layout for detecting cast shadows on the ground. Applying our shadow detector on a complex input image (a) yields false detections in the vertical structures because of complex effects like occlusion boundaries, self-shadowing, etc. (b) & (c). Recent work in scene layout extraction from single images [18] can be used to estimate the location of the ground pixels (d). We show how we can combine scene layout information with our shadow contour classifier to automatically detect cast shadows on the ground (e).

4 Incorporating Scene Layout

Until now, we have been considering the problem of detecting cast shadow boundaries on the ground with a classifier trained on local features and a CRF formulation which defines pairwise constraints across neighboring boundaries. While both approaches provide good classification accuracy, we show in Fig. 3 that applying them on the entire image generates false positives in the vertical structures of the scene. Reflections, transparency, occlusion boundaries, self-shadowing, and complex geometry [30] are common phenomena that can confuse our classifier. This results in image-wide classification results which might not be useful for complex scenes (see Figs. 3(b)-(c)).

The advent of recent approaches which estimate a qualitative layout of the scene from a single image (e.g. splitting an image into three main geometric classes: the sky, vertical surfaces, and ground [18]) may provide explicit knowledge of where the ground is. Since such a scene layout estimator is specifically trained on general features of the scene and not the shadows, combining its output with our shadow detector should reduce the number of false positive (non-shadow) detections outside the ground. We now consider how such high-level scene reasoning can be used within our shadow detection framework.

4.1 Combining Scene Layout with Local Shadow Cues

To combine the scene layout probabilities with our local shadow classifier, we can marginalize the probability of shadows over the three geometric classes sky $\mathcal{S}$, ground $\mathcal{G}$, and vertical surfaces $\mathcal{V}$:

$$P_{comb}(y_i|\mathbf{x}) = \sum_{c\in\{\mathcal{G},\mathcal{V},\mathcal{S}\}} P(y_i|c_i,\mathbf{x}_i)P(c_i|\mathbf{x}_i) \ , \tag{4}$$

where c_i is the geometric class label of boundary i, $P(y_i|c_i,\mathbf{x}_i)$ is given by our local shadow classifier, and $P(c_i|\mathbf{x}_i)$ by the scene layout classifier (we use the geometric context algorithm [18]). Unfortunately, this approach does not actually improve classification results because while it gets rid of false positives in the vertical structures, it also loses true positives on the ground along the way. This is due to the fact that shadow likelihoods get down-weighted by low-confidence ground likelihoods. Thus, we need a different approach.

4.2 Combining Scene Layout with Shadow Contours

Intuitively, we would like to penalize an assignment to the shadow class when the probability of being on the ground is low. When it is high, however, we should let the shadow classifier decide. We can encode this behavior simply by modifying the unary potentials $\phi_i(y_i)$ from (2) in our CRF formulation:

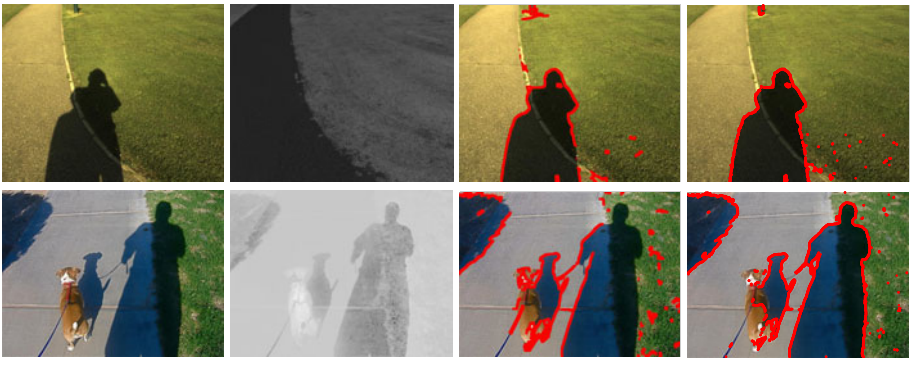
$$\phi_i(y_i) = \begin{cases} -\log P(c_i=\mathcal{G}|\mathbf{x}_i) - \log P(y_i=1|\mathbf{x}_i) & \text{if } y_i=1 \text{ (shadow)} \\ (1-P(c_i=\mathcal{G}|\mathbf{x}_i)) - \log P(y_i=0|\mathbf{x}_i) & \text{if } y_i=0 \text{ (non-shadow)} \end{cases} . \tag{5}$$

Here, $\lambda = 0.5$ and $\beta = 16$ was found by cross-validation. They yield a good compromise between local evidence and smoothness constraints.

This approach effectively combines local and mid-level shadow cues with high-level scene interpretation results, and yields an overall classification accuracy of 84.8% on our test set (see Fig. 5) without adding to the complexity of training our model. Observe how the results are significantly improved in Fig. 3(e) as compared to the other scenarios in Fig. 3(b)-(c).

5 Experimental Results

We evaluate our approach on 135 consumer photographs downloaded from LabelMe [16], Flickr, and images from the dataset introduced in [19]. In all cases, we have no control over the acquisition settings, so images contain the typical sources of distortions [31] such as jpeg compression, sharpening, sampling due to resizing, non-linear response functions, image noise, etc. We first compare our method with the current state of the art [12], then show shadow detection and removal results on several challenging images.



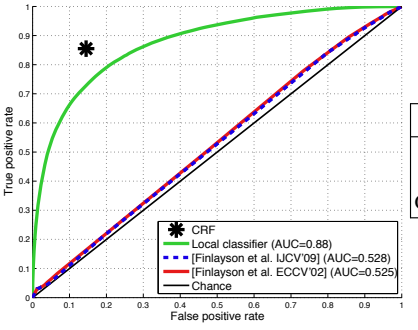
(a) Original image (b) Invariant img. [12] (c) Result with [12] (d) Our result

Fig. 4. Comparison with the shadow detection method of Finlayson et al. [12]. *First row*: Using a high-quality linear image as input (a), our implementation of their method successfully recovers a shadow-free 1-D invariant image (b), which is used to detect shadow edges (c). *Second row*: However, if the input is not linear and corrupted by noise or jpeg compression typical of consumer photographs, the 1-D invariant image still contains some shadows (b), making it hard to tell them apart from other types of edges (c). Our method detects shadows both in high and low quality images (d).

5.1 Comparison with Previous Work

The current state of the art in color-based shadow detection and removal in single images is the approach of Finlayson et al. [12], which relies on a physics-based model of shadows to compute an illumination-invariant image. Shadows are then obtained by finding edges in the original image which are not present in the invariant image. The first row of Fig. 4 shows that our implementation of their approach successfully recovers a shadow-free invariant image from the same high-quality, linear image used in their original paper [10]. Note that our approach is able to extract similar results to theirs (Fig. 4(d)). When applying their method on an image from our set, the performance degrades because the invariant still contains strong traces of shadows (second row of Fig. 4).

This is also demonstrated in the quantitative comparison shown in Fig. 5-(a), which compares our approach with the original shadow detection technique of [11], and the most recent version [12]. Note here that we cannot generate an ROC curve with the results of our CRF formulation, since the output is binary, so we plot a point at its classification accuracy. To obtain ROC scores for the competing methods, we first estimate the invariant image, and compute the difference of gradient magnitudes between the original and the invariant images. For fairness, we evaluate this score only at the strong boundaries in the image. The ROC curve shows that our results greatly outperform the previous work. This is most likely due to the use of features which are robust to artifacts common in consumer photographs.



(a) ROC curve comparison with previous work

	Shadows	Non-shadows	Combined
Local	78.3%	81.0%	79.7%
CRF	78.7%	82.3%	80.5%
CRF + scene layout	73.1%	96.4%	**84.8%**

(b) Quantitative ground shadow classification results

Fig. 5. Quantitative results. We compare our local classifier with the methods proposed by Finlayson et al. [11,12] (a). The table in (b) show the results obtained with the approaches presented in Sects. 2 (local), 3 (CRF) and 4 (CRF + scene layout). Integrating scene layout information from [18] results in ground shadow classification accuracy of 84.8%.

5.2 Ground Shadow Detection and Removal

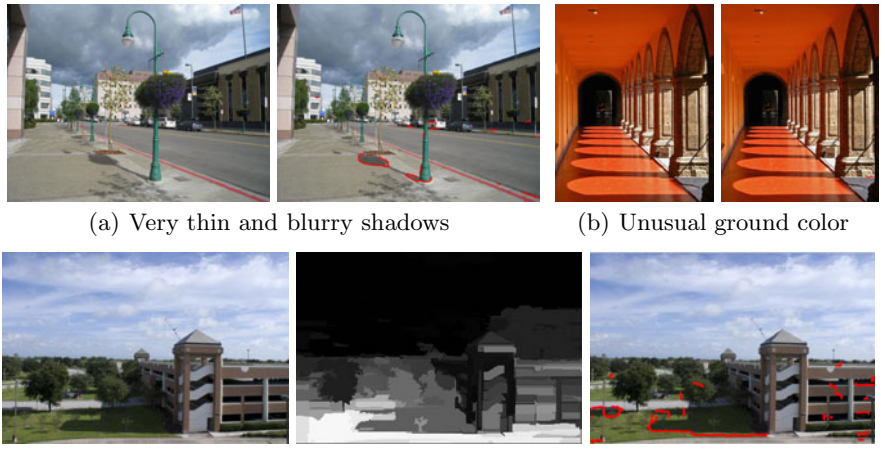
We summarize the quantitative results obtained by the technique presented in this paper on our test set in Fig. 5-(b), which shows results obtained on the entire image by the local classifier and the boundary CRF (Sects. 2 and 3), and those obtained by combining the geometric context ground likelihoods (Sect. 4). The best performance (84.8% accuracy) is obtained by our CRF formulation which combines the scene layout results with our local shadow boundary classifier.

Fig. 6 shows ground shadow detection results on several images from our dataset. It demonstrates that our method works on challenging outdoor images with varying illumination conditions, ground colors and textures, and clutter.

The typical errors made by our method are shown in Fig. 7. It may fail to detect shadows cast by thin structures like the lamppost in Fig. 7(a). Another failure case arises when the ground has a color that is vastly different from all the other images in the training set, as in Fig. 7(b). This can likely be improved by increasing the size and diversity of the training set. A third failure mode is due to errors in the estimated scene layout probabilities as in Fig. 7(c).

Once we have detected shadow boundaries, we can, as an application, use the technique introduced in [10] to remove them and recover a shadow-free image. There have been improvements proposed since then [32], but we chose the original method for its simplicity. This approach involves setting the derivatives of the image at shadow boundaries to zero, and reintegrating the result by solving the Poisson equation with Neumann boundary conditions. Fig. 8 shows shadow-free images that were computed using the boundaries detected by our method.

Fig. 6. Ground shadow detection results on images downloaded from the web (Flickr, LabelMe [16]), and the dataset from [19]. First and third columns: input images; second and fourth columns: detected ground shadows. Our approach successfully detects ground shadows in many challenging, real-world conditions.



(a) Very thin and blurry shadows (b) Unusual ground color

(c) Ground estimation errors

Fig. 7. Failure cases. The downside of using boundaries from an over-segmentation is the trade-off between spatial support obtained from longer boundaries and the size of shadow regions that can be detected. In our current setting, it may miss thin and blurry edges, like the lamppost in (a). Our approach is also sensitive to vastly different ground colors, which have never been seen by the classifier (b). Although our ratio-based features are somewhat color independent, they are not able to compensate for such drastic differences. Increasing the variety of ground colors in the training set would likely improve performance on such extreme cases. (c) Errors in the scene layout probabilities can lead to false positives.

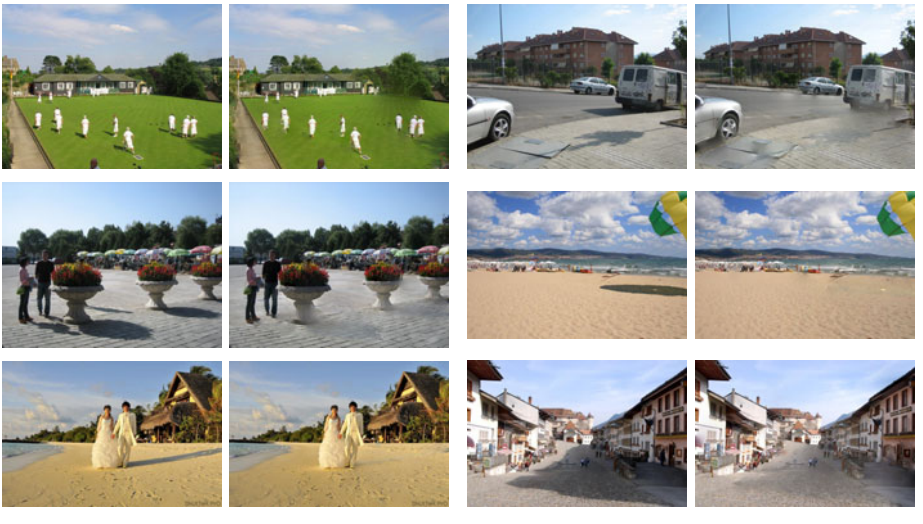


Fig. 8. Automatic ground shadow removal from a single image. We apply the original gradient reintegration method of [10] on the boundaries detected by our method. The shadows are either completely removed or greatly attenuated, with few visual artifacts.

5.3 Future Work

While our technique detects ground shadows with good accuracy, shadows that are not on the ground exhibit significantly larger appearance variations, so detecting them will be challenging. While our technique can be used as a standalone shadow detector, we believe it can also be tightly integrated into higher level scene understanding tasks. For example, the presence of an object implies that a shadow should be nearby, and vice versa. We will be pursuing these research avenues as future work.

Acknowledgements

This work has been partially supported by a Microsoft Research Graduate Fellowship to J.-F. Lalonde, an Okawa Foundation Grant, ONR grants N00014-08-1-0330 and DURIP N00014-06-1-0762, as well as NSF CAREER IIS-0643628 and IIS-0546547. We would like to thank Andrew Stein, Derek Hoiem, and Olga Veksler for making their code available online.

References

1. Lalonde, J.F., Efros, A.A., Narasimhan, S.G.: Illumination estimation from a single outdoor image. In: IEEE International Conference on Computer Vision (2009)
2. Sato, I., Sato, Y., Ikeuchi, K.: Illumination from shadows. IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2003)
3. Matsushita, Y., Nishino, K., Ikeuchi, K., Sakauchi, M.: Illumination normalization with time-dependent intrinsic images for video surveillance. IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004)
4. Finlayson, G.D., Fredembach, C., Drew, M.S.: Detecting illumination in images. In: IEEE International Conference on Computer Vision (2007)
5. Weiss, Y.: Deriving intrinsic images from image sequences. In: IEEE International Conference on Computer Vision (2001)
6. Huerta, I., Holte, M., Moeslund, T., Gonzàlez, J.: Detection and removal of chromatic moving shadows in surveillance scenarios. In: IEEE International Conference on Computer Vision (2009)
7. Wu, T.P., Tang, C.K.: A bayesian approach for shadow extraction from a single image. In: IEEE International Conference on Computer Vision (2005)
8. Bousseau, A., Paris, S., Durand, F.: User-assisted intrinsic images. ACM Transactions on Graphics (SIGGRAPH Asia 2009) 28 (2009)
9. Shor, Y., Lischinski, D.: The shadow meets the mask: pyramid-based shadow removal. Computer Graphics Forum Journal (Eurographics 2008) 27 (2008)
10. Finlayson, G.D., Hordley, S.D., Drew, M.S.: Removing shadows from images. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 146–160. Springer, Heidelberg (2002)
11. Finlayson, G.D., Drew, M.S., Lu, C.: Intrinsic images by entropy minimization. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 582–595. Springer, Heidelberg (2004)
12. Finlayson, G.D., Drew, M.S., Lu, C.: Entropy minimization for shadow removal. International Journal of Computer Vision 85 (2009)

13. Maxwell, B.A., Friedhoff, R.M., Smith, C.A.: A bi-illuminant dichromatic reflection model for understanding images. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
14. Tian, J., Sun, J., Tang, Y.: Tricolor attenuation model for shadow detection. IEEE Transactions on Image Processing 18 (2009)
15. Narasimhan, S.G., Ramesh, V., Nayar, S.K.: A class of photometric invariants: Separating material from shape and illumination. In: IEEE International Conference on Computer Vision (2005)
16. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: LabelMe: a database and web-based tool for image annotation. International Journal of Computer Vision 77 (2008)
17. Freeman, W.T., Pasztor, E.C., Carmichael, O.T.: Learning low-level vision. International Journal of Computer Vision 40 (2000)
18. Hoiem, D., Efros, A.A., Hebert, M.: Recovering surface layout from an image. International Journal of Computer Vision 75 (2007)
19. Zhu, J., Samuel, K.G.G., Masood, S.Z., Tappen, M.F.: Learning to recognize shadows in monochromatic natural images. In: IEEE Conference on Computer Vision and Pattern Recognition (2010)
20. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proceedings of the 6th International Conference on Computer Vision (1998)
21. Barnard, K., Finlayson, G.D.: Shadow identification using colour ratios. In: Proc. IS&T/SID 8th Color Imaging Conf. Color Science, Systems and Applications (2000)
22. Khan, E.A., Reinhard, E.: Evaluation of color spaces for edge classification in outdoor scenes. In: IEEE International Conference on Image Processing (2005)
23. Chong, H.Y., Gortler, S.J., Zickler, T.: A perception-based color space for illumination-invariant image processing. ACM Transactions on Graphics, SIGGRAPH 2008 (2008)
24. Collins, M., Shapire, R., Singer, Y.: Logistic regression, adaboost and Bregman distances. Machine Learning 48 (2002)
25. Hoiem, D., Stein, A., Efros, A.A., Hebert, M.: Recovering occlusion boundaries from a single image. In: IEEE International Conference on Computer Vision (2007)
26. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In: IEEE International Conference on Computer Vision (2001)
27. Boykov, Y., Veksler, O., Zabih, R.: Efficient approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (2001)
28. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004)
29. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transaction on Pattern Analysis and Machine Intelligence 26 (2004)
30. Sinha, P., Adelson, E.H.: Recovering reflectance and illumination in a world of painted polyhedra. In: IEEE International Conference on Computer Vision (1993)
31. Chakrabarti, A., Scharstein, D., Zickler, T.: An empirical camera model for internet color vision. In: British Machine Vision Conference (2009)
32. Fredembach, C., Finlayson, G.D.: Simple shadow removal. In: International Conference on Pattern Recognition (2006)

The Semi-explicit Shape Model for Multi-object Detection and Classification*

Simon Polak and Amnon Shashua

School of Computer Science and Engineering
The Hebrew University of Jerusalem

Abstract. We propose a model for classification and detection of object classes where the number of classes may be large and where multiple instances of object classes may be present in an image. The algorithm combines a bottom-up, low-level, procedure of a bag-of-words naive Bayes phase for winnowing out unlikely object classes with a high-level procedure for detection and classification. The high-level process is a hybrid of a voting method where votes are filtered using beliefs computed by a class-specific graphical model. In that sense, shape is both explicit (determining the voting pattern) and implicit (each object part votes independently) — hence the term "semi-explicit shape model".

1 Introduction

One of the great challenges facing visual recognition is scalability in the face of large numbers of object classes and detected instances of objects in a single image. The task requires both *classification*, i.e., determine if there is a class instance in the image, and *detection* where one is required to localize all the class instances in the image. The scenario of interest is where a class instance occupies a relatively small part of the image surrounded by clutter and other instances (of the same class and other classes), and all of that in the face of a large number of classes, say hundreds or thousands.

The two leading approaches for detecting multiple instances of an object class in an image are sliding windows (cf. [1,2,3]), and voting methods (cf. [4,5]), which are based on modeling the probabilities for relative locations of object parts to the object center or more generally to the Hough transform.

The sliding-window approach applies the state-of-the-art binary ("one versus many") classification in a piece-meal fashion systematically over all positions, scale and aspect ratio. The computational complexity of this scheme is unwieldy although various techniques have been proposed to deal with this issue where the most notable is the cascaded evaluation [1,6] where each stage employs a more powerful (and expensive) classifier. Controlling the false positive rate, given the very large number of classification attempts per image, places considerable challenges on the required accuracy of the classifier and is typically dealt by means of post-processing such as non-maximal suppression.

* This work was partially funded by ISF grant 519/09.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 336–349, 2010.

In contrast to this, the voting approach parametrizes the object hypothesis (typically, the location of the object center) and lets each local part vote for a point in hypothesize space. These part-based methods combine large numbers of local features in a single model by establishing statistical dependencies between parts and the object hypothesis, i.e., by modeling the probabilities for relative locations of parts to the object center [4]. In some cases, the spatial relationship among the parts are not modeled thereby modeling the object as a "bag of parts" as in the Implicit Shape Model (ISM) of [4] and in other cases shape is represented by the mutual position of its parts through a joint probability distribution [7,8,9,10]. The ISM approach is efficient and is designed to handle multiple instances of an object class, however, the lack of shape modeling contaminates the voting map with multiple spurious local maxima [5]. The probabilistic models on the other hand require a daunting learning phase of fitting parameters to complex probabilistic models although various techniques have been proposed to deal with the complexity issue such as identifying "landmark" parts [9,10] or Tree-based part connectivity graphs [8]. Moreover, the probabilistic models lack the natural ability to handle multiple instances in parallel (like ISM does), although in some cases authors [8] propose detecting multiple instances in a sequential manner starting from the "strongest" detected model after which nearby parts are excluded to find the best remaining instance and so on. Finally, both ISM and the explicit shape models would be challenged with increasing number of object classes as there is no built-in filters for winnowing out the less likely object classes given the image features before the more expensive object-class by object-class procedures are applied.

Our proposed model combines a bottom-up "bag of parts" procedure using a naive Bayes assumption with a top-down probabilistic model (per object class). The probabilistic model, on one hand, represents the shape by interconnection of its parts and uses approximate inference over a loopy graphical model to make inference. However, the inference results are not used explicitly to match a model to an image but *implicitly* to filter out the spurious votes in the ISM procedure. The voting of parts to object centers are constrained by the marginal probabilities computed from the graphical model representing the object shape. Therefore, spurious parts not supported by neighboring parts according to the shape graph would not vote. Furthermore, the locations of maximal votes are associated with a classification score based on the graphical model rather than by the amount of votes. Because shape is used both explicitly and implicitly in our model we refer to the scheme as "semi explicit shape model".

2 The Semi-explicit Shape Model

Let $C_1, ..., C_n$ stand for the n object categories/classes we wish to detect and locate in novel images. Let $P(C_k)$ be the prior on class C_k which can be estimated from the training set (number of images we have from C_k divided by the size of the training set). We assume that for each class we have a set of training images where the object is marked by a surrounding bounding box. We describe

below the training phase which consists of creating a code-book of features, defining object "parts" and their probabilistic relation to code words, and the construction of Part connectivity graph per object class. Following the training phase we describe in Section 2.2 the details of our classification and detection algorithm.

2.1 The Training Phase

We start the training phase by constructing a "code book" $\mathcal{W}$ by clustering all the descriptors gathered around all interest-points from all the training images. From the training images of the k'th object class we perform the following preparations: (i) delineate the Parts of the object each consisting of a 2D Gaussian model and the collection of interest points and their descriptors associated with the Part, (ii) a Part neighborhood graph which would serve during the visual recognition phase as a graphical model for enforcing global spatial consistency among the various Parts of the object, and (iii) construct the probabilistic representation of object Parts by the conditional likelihood $P(R \mid w)$ for all $w \in \mathcal{W}$. We present each step in more details below.

The Code Book: all training images are passed through a difference of Gaussians interest point locator and a SIFT [11] shape descriptor vector is generated per interest point and per scale. The area under each bounding box is represented at different scales and recorded with each descriptor. We use an agglomerative clustering algorithm (such as the Average-Link in [12]) to group together descriptors of similar shape and of the same scale. An agglomerative clustering bounds the quantization error (which in turn is bounded by the threshold distance parameter between descriptors) and allows to represent isolated descriptors (such as those generated by object-specific image fragments) as clusters. A K-means clustering approach, although superior computational-wise, would force isolated descriptors to get associated with some larger cluster of common descriptors, thereby increasing the quantization error. The i'th cluster is denoted by w_i and consists of the descriptor vectors $d_{i_1}, ..., d_{i_{m_i}}$ and the average descriptor d_i where m_i is the cluster size. Each code word is associated with some scale (as the clustering is performed for each scale separately). The code-book $\mathcal{W}$ is the set of "code words" $w_i(s)$, $i = 1, ..., M$ and s is the scale label.

Object Parts Delineation: we define an object "part" by a concentration of interest points, collected over all the training images of the class. We do not require the interest points to share similar descriptors in order to allow for appearance variability within the scope of the Part. For example, the area surrounding the Eye in a frontal human face is a natural part, yet people wear glasses which renders the appearance of that area in the image undergo considerable variation. On the other hand, our working assumption is that concentrations of interest-points undergo only moderate variability. Thus, radically different viewing positions of an object, for example, are not currently included in our model of an "object class". The point concentrations are detected and modeled as

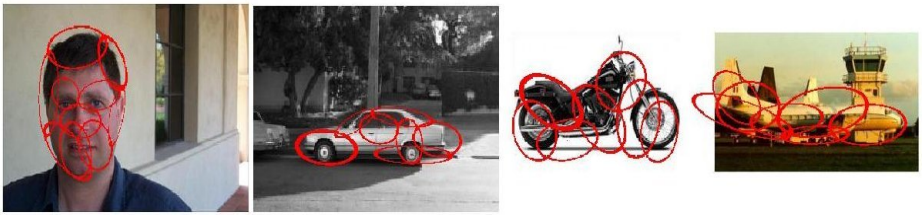


Fig. 1. Examples of model Parts for some classes of Caltech101 database. Each ellipse depicts a 2D Gaussian associated with a separate Part.

follows. Given all the training images of class C_k, the bounding boxes around the object are scale-aligned and interest point locations are measured relative to the bounding-box center (object center). The collected interest-points over all the training images of C_k are fed into a Gaussian-Mixture-Model (GMM) using the Expectation-Maximization algorithm [13]. The number of Parts (Gaussian models) is determined by a minimum-description-length principle described in [14]. The result is a list of Parts R_j^k represented by $N(\mu_j^k, \Sigma_j^k)$ a 2D Gaussian model, for $j = 1, ..., n_k$ where n_k is the number of Parts of object class C_k. Note that we have tacitly assumed that scale does not influence the Part structure of the object (number and shape distribution). The assumption holds well in practice under a large range of scales and simplifies the algorithm. Fig. 1 illustrates the Parts found in some of the Caltech101 images.

We define for each class a "context" Part R_B^k which consists of the set of descriptors from interest points located in the *vicinity* of the object bounding box and collected over all the training images of C_k. The Context Part will be used in the next section as additional evidence for the likelihood of C_k given a novel image.

In addition, let F_j^k be the set of descriptors of the interest points which were assigned by the GMM algorithm to Part R_j^k. Since GMM provides a probabilistic assignment of interest points to Parts, each interest point can belong to more than one Part. We leave only the strong (above threshold) assignments, i.e., each interest point is associated with the highest probability Parts. Finally, let $F^k = \bigcup_j F_j^k$ stand for the set of all descriptors of interest points of class C_k, and $F = \bigcup_k F^k$ the set of all descriptors collected from the training set.

Probabilistic Representation of Parts $P(R_j^k \mid w_i)$**:** we wish to represent the Part R_j^k by its conditional probability given a word w_i. Such a representation is useful for determining the likelihood of having R_j^k in an image given interest points and their SIFT descriptors which in turn can be used to obtain a preliminary classification score based on a naive Bayes model.

To compute $P(R_j^k \mid w_i)$, let $|F_j^k \cap w_i|$ denote the number of descriptors that are in both the part R_j^k and the code word w_i. The ratio $|F_j^k \cap w_i|/|w_i|$ is not a good representation of $P(R_j^k \mid w_i)$ because it makes a tacit assumption that the

prior $P(C_k)$ is equal to $|F_k|/|F|$ the relative number of descriptors from C_k — an assumption that is obviously wrong.

We expand $P(R_j^k \mid w_i)$ while noting that $P(R_j^{k'} \mid C_k) = 0$ when $k' \neq k$:

$$\begin{aligned} P(R_j^k \mid w_i) &= P(R_j^k \mid C_k, w_i) P(C_k \mid w_i) \\ &= P(R_j^k \mid C_k, w_i) \frac{P(w_i \mid C_k) P(C_k)}{P(w_i)} \\ &= \frac{|F_j^k \cap w_i|}{|F^k \cap w_i|} \frac{\frac{|F^k \cap w_i|}{|F^k|} P(C_k)}{|w_i|/|F|} \end{aligned}$$

Note that if we substitute $|F_k|/|F|$ for $P(C_k)$ we obtain the ratio $|F_j^k \cap w_i|/|w_i|$. Following the cancelation of the term $|F^k \cap w_i|$ we obtain:

$$P(R_j^k \mid w_i) = \frac{|F_j^k \cap w_i| \cdot |F| \cdot P(C_k)}{|F_k| \cdot |w_i|} \tag{1}$$

Note that the definition above applies to $P(R_B^k \mid w_i)$ as well where F_j^k is replaced by F_B^k the set of descriptors of the Context Part.

Constructing the Part Connectivity Graph: an explicit shape model of class C_k is represented by a connected (undirected) graph $G(V^k, E^k)$ whose set of nodes V^k correspond to the Parts R_j^k, $j = 1, ..., n_k$ and whose set of edges E^k defines the "Part neighborhood" to guarantee a global consistency structure among the Parts. The neighborhood relations are determined by a Delaunay triangulation [15] over the Gaussian centers μ_j^k which form the Part centers.

2.2 Detection and Recognition of Object(s) Instances in a Novel Image

The training phase described above has generated (i) a code book $\mathcal{W}$ where each word $w(s) \in \mathcal{W}$ represents a set of image descriptors of similar appearance and scale s, (ii) the j'th object Part R_j^k of class C_k represented by a 2D Normal distribution in object-centered coordinates, (iii) a "bag of words" association between object Parts R_j^k and code words w_i represented by the scalar $P(R_j^k \mid w_i)$ (eqn. 1), and (iv) a Part connectivity graph.

Given a novel image I we wish to detect and recognize instances of the object classes $C_1, ..., C_k$ allowing for multiplicity of objects and multiplicity of instances of each object at different scales. The detection and classification process has two phases:

- A low-level, bottom-up, "bag of words" based classification of object classes. Classification is based on the association $P(R_j^k \mid w_i)$ over all code-words and Parts of each object class. Classification also forms a ranking of the possible object classes thereby allowing the system to focus its high-level resources on the most likely object classes that may be present in the image first.

– A high-level classification and detection process: for each of the likely classes C_k, the Part connectivity graph is matched to the image using a Tree-Reweighted (TRW) approximate inference over a loopy graphical model. Each Part obtains "beliefs" on its possible locations in the image (allowing for multiple instances). The Part locations with high Belief vote for the respective object-class center. The result is a "heat map" (like with the ISM method) of possible centers of instances from C_k. Each object-center candidate in the heat-map is associated with a score given by the graphical model inference which serves as a high-level classification score. This high-level process is performed sequentially over each object-class limited to those classes with high likelihood (as determined by the low-level phase).

We describe the two phases in detail below.

Likelihood of Classes as a Low-Level Process: the low-level classification process is triggered from detected interest points and their associated SIFT descriptors from the novel image. A nearest-neighbor search is performed to match the descriptor of each interest point to a code-word. Because of the relatively high dimension of the SIFT descriptor we use the locally-sensitive-hashing (LSH) method based on random projections [16]. Let w_I be the subset of code words present in the input image, then the conditional likelihood $P(R_j^k \mid I)$ of the Part R_j^k existing in novel image I is:

$$P(R_j^k \mid I) = \sum_{w_i \in w_I} P(R_j^k \mid w_i) P(w_i \mid I),$$

and the conditional log-likelihood $\log P(C_k \mid I)$ of the class C_k given the novel image is determined by a Naive Bayes approach:

$$\log P(C_k \mid I) = \sum_{j=1}^{n_k} \log P(R_j^k \mid I) + \log P(R_B^k \mid I), \tag{2}$$

where R_B^k is the Context part (defined above). The probabilistic representations above are "bag of words" type of inference where the likelihoods of Parts and object classes depend only on the existence of features (code words) and not through their spatial interconnection. The inference of $\log P(C_k \mid I)$ follows from a Naive-Bayes assumption on a co-occurrence relation between objects and parts. This "weak" form of inference is efficient and allows us to perform a preliminary classification which also serves as a ranking of the possible classes by means of $\log P(C_k \mid I)$. A similar approach of using nearest-neighbors with a naive-Bayes approach (but without a code book and other details of Parts and their probabilistic relation to code words) was introduced by [17].

High-level Classification and Detection: this phase is performed on each object class C_k whose classification score $\log P(C_k \mid I)$ was above threshold, i.e., the high-level process focuses its resources on the most likely object classes first. We

construct an inference problem defined by a joint probability $P(x_1^k, ..., x_{n_k}^k)$ using the connectivity graph $G(V^k, E^k)$ for defining direct interactions among the variables. The variable x_j^k is defined over a finite set of values representing the possible locations of the Part R_j^k in the image. The marginal probability distribution $P(x_j^k)$ represents the probability ("belief") $P(x_j^k = r)$ for R_j^k to be found in location r in the image. Each possible location r votes to C_k's object center if $P(x_j^k = r)$ is above threshold. The result of the voting process is a "heat-map" for instances of C_k in the image. The value of $P(x_1^k = r_1, ..., x_{n_k}^k = r_{n_k})$ provides a classification score of an instance of C_k at a specific location in the image where, unlike the low-level phase where the score was based on a "bag-of-words" setting, the score is based on satisfying the connectivity constraints among object parts. We therefore have both detection (via the heat-map) and classification achieved simultaneously. We present the scheme in more details as follows.

Let $\mathcal{I} = I_1, ..., I_M$ be the set of interest points and their associated descriptors located in the novel image and let $w_1, ..., w_M$ the corresponding code-words (found using LHS nearest-neighbor approximation). Let $I_j^k \in \mathcal{I}$ be the subset of interest points for which their corresponding code-words w_i satisfy $P(R_j^k \mid w_i) > \epsilon$ for some threshold ϵ. In other words, the set I_j^k are the interest points in the novel image that are likely to belong to the Part R_j^k. We perform agglomerative clustering on I_j^k where the similarity measure is the Mahalanobis distance with zero mean and covariance matrix of R_j^k (recall that each Part is associated with a Normal distribution) for each pair arising from the same scale and infinity otherwise. Since each code word has an associated scale, interest points arising from different scales will not be clustered together. Let n_j^k be the number of clusters found and $\gamma_1, ..., \gamma_{n_j^k}$ are the clusters of the respective code words associated with I_j^k and $l_1, ..., l_{n_j^k}$ are the geometric centers of the clusters. Let $x_j^k \in \{1, ..., n_j^k\}$ be a random variable associated with the possible locations of the Part R_j^k (where each location is a cluster of interest points of scale s for which $P(R_j^k \mid w_i(s)) > \epsilon$).

The joint probability distribution over the variables x_j^k, $j = 1, ..., n_k$ has the form:

$$P(x_1^k, ..., x_{n_k}^k) = \frac{1}{Z} \prod_{j=1}^{n_k} \phi_j(x_j^k) \prod_{(i,j) \in E^k} \psi_{i,j}(x_i^k, x_j^k), \tag{3}$$

where $\phi_j(x_j^k)$ represents the "local evidence", i.e., $\phi_j(x_j^k = r)$ is the probability that R_j^k is located at location r from local evidence alone:

$$\phi_j(x_j^k = r) = 1 - \prod_{w_i \in \gamma_r} \left[1 - P(R_j^k \mid w_i)\right],$$

and $\psi_{i,j}(x_i^k, x_j^k)$ are the pairwise "potential" functions on pairs of Parts that are incident in the connectivity graph. The value of $\psi_{i,j}(x_i^k = r, x_j^k = q)$ represents the likelihood that the two Parts are located in positions r, q (and scale s) respectively:

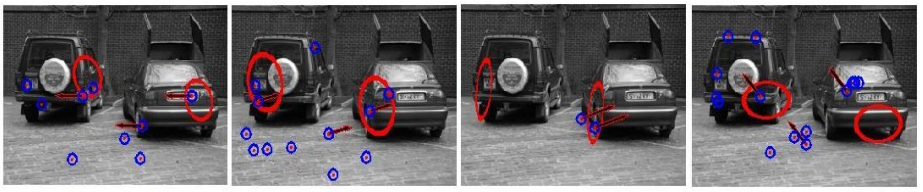


Fig. 2. Each image shows a Part R_j^k (Red Ellipse) with the set of candidate locations x_j^k. Locations with high belief are those who vote and are drawn with an arrow pointing to the object center. The beliefs generated by the graphical model form a strong constraint on the voting pattern of Part candidates so that only those locations who have global shape support end up voting. The images contain multiple instances thus the belief pattern of $P(x_j^k)$ is multi-modal. Candidate locations from both object instances end up voting.

$$\psi_{i,j}(x_i^k = r, x_j^k = q) = N(l_r - l_q; \mu_{ij}, \Sigma_{ij}),$$

where μ_{ij}, Σ_{ij} is the scaled difference Normal distribution where $\mu_{ij} = (\mu_i^k - \mu_j^k)s$ and $\Sigma_{ij} = (\Sigma_i^k + \Sigma_j^k)s^2$. We set $\psi() = 0$ in case positions r, q are associated with different scales.

The marginal probabilities $P(x_j^k)$ hold the likely Part locations, i.e., if $P(x_j^k = r)$ is above threshold then we have a certain "belief" that l_r (the geometric center of γ_r the r'th cluster) is where the Part R_j^k is centered. Because we may have multiple instances of C_k in the image, $P(x_j^k = r)$ may have a multi-modal profile where more than a single Part location is supported by the connectivity graph.

Computing the marginal probabilities is computationally infeasible and instead we resort to "approximate inference". Since the connected graph has loops, the sum-product Belief-Propagation (BP) algorithm is not guaranteed to converge. Moreover, regardless of convergence, the BP algorithm tends to settle on single-modal beliefs, i.e., $P(x_j^k)$ will come out single-modal even when multiple instances of C_k exist in the image. We used the Tree-reweighted (TRW) convex-free-energy variational approximation which is both guaranteed to converge and is not limited to single-modal solutions. Specifically, we used the sum-TRBP [18] implementation (even though convergence is not guaranteed). Convergence guaranteed TRW algorithms (and general convex-free-energy) can be found in [19].

The marginal probabilities $P(x_j^k)$ play two roles in the high-level detection and classification process. First is to "clean up" the voting of Part candidates to object centers, and second to obtain a high-level (shape-based) classification score for each detected instance of C_k in the image. Those are detailed below.

Voting: once the (approximate) marginal probabilities $P(x_j^k)$ are estimated we perform a voting procedure: For each Part R_j^k, the candidate Part centers l_r will vote to the respective object center if $P(x_j^k = r)$ is above threshold. Fig. 2 illustrate the constrained voting procedure: in each image a Part is shown marked by an Ellipse and all candidate locations for the Part are marked by circles. Only

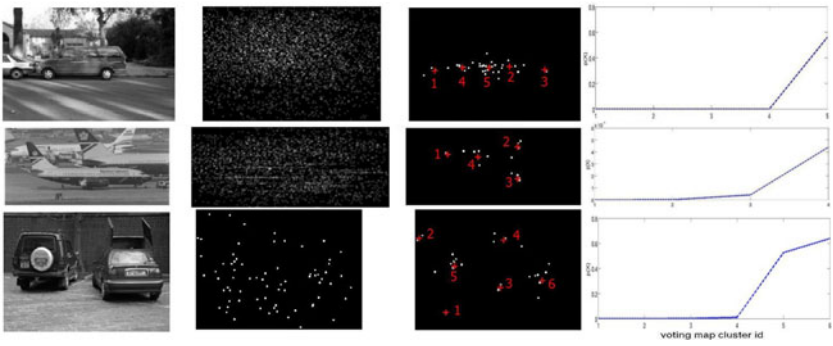


Fig. 3. From heat-map to classification score: the middle column shows the heat map generated by ISM (i.e., without our high-level filtering using beliefs generated from sum-TRBP). The third column shows the heat-map generated by our algorithm. It is evident that most of the voting contamination has been removed. The centers of maximal votes found by Mean-Shift are marked on the heat-maps. The righthand column shows the classification score (generated by the joint probability distribution) associated with each of the heat map centers. The top and bottom rows show the cases where the class is the correct one and one can see that the true heat map center has the (significantly) highest classification score (No. 5 in top, and 5,6 in bottom). The middle row shows a case where the class is not found in the image. In that case all classification scores are close to zero (the scale is 10^{-3}).

those locations which received high belief make a vote and are displayed with an arrow towards the object center. It is evident than only a small fraction of the possible locations eventually make a vote and that the procedure is able to concentrate on both instances simultaneously due to the usage of the sum-TRBP algorithm.

In other words, the voting process is a "filtered" version of the ISM method. Rather than having all Part candidates vote for their respective object center, only those candidates with high Belief perform the voting. This "high-level filter" has a dramatic effect on reducing the "clutter" formed by spurious votes on the resulting object-centers "heat map" (see Fig. 3).

High-level Classification: the voting process creates a heat-map where locations having many votes are likely to form centers of instances of C_k, thus the "strength" of a candidate instance can be directly tied to the number of votes the center has received — this is the underlying premise of ISM. However, we can do better and obtain a classification measure by evaluating $P(x_1^k, ..., x_{n_k}^k)$ for every instance candidate (a center receiving sufficient votes), as follows. Consider a candidate center c and the set of locations $\mathcal{L}_c$ which have voted to it. Each location is associated with a Part R_j^k and with a value of its corresponding position label x_j^k. Let $\mathcal{L}_c(j,k) \subset \mathcal{L}_c$ be the locations corresponding to R_j^k and let $r_1, ..., r_b$ be the values of x_j^k corresponding to the locations $\mathcal{L}_c(j,k)$. Normally $b = 1$, i.e., there is only one location for R_j^k and the value of x_j^k is set accordingly

(to r_1). In case $b > 1$, then $x_j^k = \text{argmax}_q P(x_j^k = r_q)$. In case $\mathcal{L}_c(j,k) = \emptyset$, i.e., Part R_j^k did not vote to center c, then x_j^k is set to the label with maximal belief. Once $x_1^k, ..., x_{n_k}^k$ are set to their value, we evaluate $P(x_1^k, ..., x_{n_k}^k)$ according to eqn. 3. The value of the joint probability measure both local fit of Parts and global consistency among parts and therefore serves as our classification score of the candidate instance of C_k at center c. The difference between the Naive-bayes score $P(C_k \mid I)$ (eqn. 2) and the high-level classification score is dramatic at time boosting accuracy of recognition by significant amounts. Fig. 3 shows examples of heat-maps with the maximal centers (estimated using mean-shift procedure) together with the classification scores associated with those centers. It is evident that true center candidates have a much higher classification score than spurious centers (despite them having a similarly large number of votes). In images where the object class is not present, all candidate centers have a low classification score.

3 Experiments

We have tested our model on two standard datasets, Clatech101 [20] and Pascal VOC 2006 [21]. The Caltech101 dataset contains images containing a single dominant object from 101 classes including cars, faces, airplanes, motorbikes among other classes. The instances from those classes appear approximately at similar scale and pose in all images. Each object class is found in between 100 to 800 images. The Pascal dataset is more challenging as it contains 5000 images, split evenly to training and testing subsets, of ten object classes with varying scale and viewpoint where each image may contain multiple instances of object classes. As a result objects are less dominant in the image compared to Caltech101 thereby making the task of detection and classification challenging. Fig. 4 shows the Parts detected in test images by taking the locations of highest belief for each part of the object class in question. One can see the detected Parts agree with their true locations on the test images.

With the Calctech101 dataset we performed the object versus other objects categorization experiment, where the goal is to classify an image to one of the 101 object classes. We have removed the *Faces_easy* class, since the objects is this class are identical to the objects in the class *Faces*, so the number of classes in our experiments was 100. In this test we selected a training set of 15 images per class and a test set of 15 images per class. We collected around $750,000$ features for each object scale (we have used 5 scales) and clustered them into a code book of sizes ranging from $60,000$ to $80,000$ and the number of Parts per object varied between 8 to 15. During the testing phase, each image produced between $100 - 1000$ interest points and each part had between $10 - 30$ possible locations. Mean running time for a test image was under 5 seconds on a standard 3GHZ CPU. We ran both classifiers: our low-level naive Bayes classifier $P(C_k \mid I)$ and the high-level detection and classification(in this case the categorization is performed by selecting the class with highest detection score). Table 1 shows comparison of our results to other methods on the Caltech101 dataset.

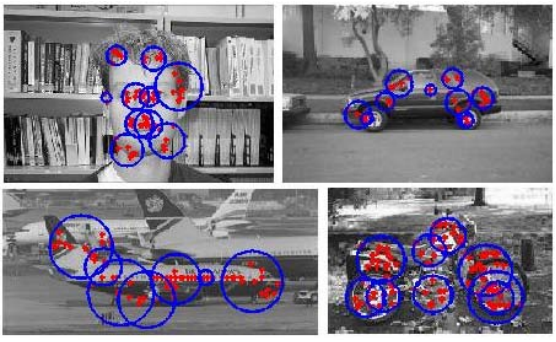


Fig. 4. Examples of correct detections of classes 'face', 'car', 'motorbikes' and 'airplanes' from Caltech101 dataset. Each circle in the images represent most probable location of a different Part of the object's shape model. The Red dots inside the circles are the interest points belonging to this Part.

Table 1. Categorization performance comparison our approach and other methods on the Caltech101 dataset

Naive Bayes	High-Level	[17]	[22]	[23]	[24]	[25]
51.70%	68.80%	65.00%	59.30%	59.10%	52.00%	56.40%

With the Pascal VOC 2006 dataset, we used the provided training set (of 2500 images) to create a model for each of the four view points of each object and tested our algorithm in both categorization and detection tasks. From the training images of the Pascal database we extracted more then 2,500,000 SIFT features, which resulted in around 100,000 code words for each scale. During the model creation we have used the view information available in the dataset to construct separate models for each of the existing four views (left, right, rear and frontal) in a similar manner to that used for Caltech101.

For the classification test, the classification score is computed (by taking the center with the highest classification score from the heat map) per object class. Since an image can contain a number of object classes, an ROC curve is constructed and the area under the curve is taken for the performance measure. Table 2 shows the classification performance of our algorithm for all the ten classes, compared to the low-level naive Bayes phase of our algorithm. In most classes the shape model boosts the performance but in some case, such as with the class of Pedestrians, the performance actually decreases. The reason for that is that Pedestrians instances are sometimes at a very small scale and the system does not detect a sufficient number of interest points to enable the graphical model to perform as expected. On the other hand, those images often contain multiple Pedestrians thus the "bag of code words" underlying the naive Bayes procedure collects evidence from the multiple instances.

For the detection task, performance is measured by the overlap between bounding boxes. Fig. 5 shows some detection results on a sample of test images

Table 2. Performance comparison between the high-level classification and the naive Bayes low-level classification on the Pascal VOC 2006 dataset

	bicycle	bus	car	cat	cow	dog	horse	motorbike	person	sheep
High-Level	90%	93%	90.9%	85.4%	88.5%	77.3%	72.4%	86.4%	60%	87.3%
Naive Bayes	87.3%	90.7%	89%	82.5%	85.9%	75.7%	68.4%	78.7%	67.7%	82.7%

Fig. 5. Examples of detections from the Pascal VOC 2006 dataset (see discussion in text)

Table 3. Performance comparison between the proposed algorithm and published results by other methods (sliding window and voting) on the Pascal VOC 2006 dataset

	bicycle	bus	car	cat	cow	dog	horse	motorbike	person	sheep
Our	0.36	0.184	0.621	0.171	0.39	0.18	0.37	0.55	0.33	0.41
Cambridge	0.249	0.138	0.254	0.151	0.149	0.118	0.091	0.178	0.030	0.131
ENSMP	-	-	0.398	-	0.159	-	-	-	-	-
INRIA Douze	0.414	0.117	0.444	-	0.212	-	-	0.390	0.164	0.251
INRIA Laptev	0.44	-	-	-	0.224	-	0.140	0.318	0.114	-
TUD	-	-	-	-	-	-	-	0.153	0.074	-
TKK	0.303	0.169	0.222	0.160	0.252	0.113	0.137	0.265	0.039	0.227
FeiFei09]	-	-	0.310	-	-	-	-	-	-	-
Felzenszwalb'09	0.619	0.49	0.615	0.188	0.407	0.151	0.392	0.576	0.363	0.404

where we can see the ability of the algorithm to handle occlusions, view and scale variations and multiple instances of an object appearing in the same image. Table 3 summarizes the detection performance of our algorithm in comparison to other methods. As it can be seen from the table our system outperforms many methods on most of the classes except the sliding-window method by [3]. The running time per image in the Pascal dataset is less than 4 seconds compared to much longer running times by other methods.

4 Summary

We described an object detection and classification scheme based on a voting mechanism. Our system starts with a bottom-up Naive-Bayes "bag of words"

classification for ranking the possible class models present in the image followed by a top-down voting of visual code words (through Parts) to potential object classes. The voting mechanism is filtered by explicit shape models represented by graphical models. The "beliefs" computed by each of the graphical models leave intact votes from code-words which gain structural support by other code-words in the graph. The system is designed to scale gracefully with the number of classes and achieves comparable, and often superior, detection and classification accuracies than other systems which have a considerably higher run-time.

References

1. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 511–518 (2001)
2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 886–893 (2005)
3. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminantly trained, multiscale, deformable part model. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2008)
4. Leibe, B., Leonardis, A., Schiele, B.: Combined object detection and segmentation with an implicit shape model. In: ECCV 2004 Workshop on Statistical Learning in Computer Vision (2004)
5. Ommer, B., Malik, J.: Multi-scale object detection by clustering lines. In: Proceedings of the International Conference on Computer Vision (2009)
6. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: Proceedings of the International Conference on Computer Vision (2009)
7. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2003)
8. Felzenszwalb, P., Huttenlocher, D.: Pictorial structures for object recognition. International Journal of Computer Vision 61, 55–79 (2005)
9. Fergus, R., Perona, P., Zisserman, A.: A sparse object category model for efficient learning and exhaustive recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2005)
10. Crandall, D., Felzenszwalb, P., Huttenlocher, D.: Spatial priors for part-based recognition using statistical models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2005)
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110 (2004)
12. Leibe, B., Mikolajczyk, K., Schiele, B.: Efficient clustering and matching for object class recognition. In: British Machine Vision Conference, BMVC 2006 (2006)
13. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Stat. Soc., Series B 39, 1–38 (1977)
14. Wallace, C.S., Freeman, P.R.: Estimation and inference by compact coding. Journal of the Royal Statistical Society. Series B (Methodological) 49, 240–265 (1987)
15. Cignoni, P., Montani, C., Scopigno, R.: Dewall: A fast divide and conquer delaunay triangulation algorithm in e^d. Computer-Aided Design 5, 333–341 (1998)

16. Gionis, A., Indyk, P., Motwani, R.: Similarity Search in High Dimensions via Hashing. In: Proceedings of the 25th Very Large Database (VLDB) Conference (1999)
17. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: CVPR (2008)
18. Wainwright, M., Jaakkola, T., Willsky, A.: A new class of upper bounds on the log partition function. IEEE Transactions on Information Theory 51, 2313–2335 (2005)
19. Hazan, T., Shashua, A.: Convergent message-passing algorithms for inference over general graphs with convex free energies. In: Conference on Uncertainty in Artifical Intelligence (UAI), Helsinki, Finland (2008)
20. 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)
21. Everingham, M., Zisserman, A., Williams, C.K.I., Van Gool, L.: The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results (2006), http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf
22. Varma, M., Ray, D.: Learning the discriminative power-invariance trade-off. In: Proceedings of the International Conference on Computer Vision (2007)
23. Zhang, H., Berg, A., Maire, M., Malik, J.: Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2006)
24. Berg, A.: Shape matching and object recognition (2005)
25. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2169–2178 (2006)

Coupled Gaussian Process Regression for Pose-Invariant Facial Expression Recognition

Ognjen Rudovic[1], Ioannis Patras[2], and Maja Pantic[1,3]

[1] Comp. Dept, Imperial College, London, UK
[2] Elec. Eng. Dept, Queen Mary University, London, UK
[3] EEMCS, University of Twente, 7500 AE Enschede, The Netherlands
{o.rudovic,m.pantic}@imperial.ac.uk, i.patras@elec.qmul.ac.uk

Abstract. We present a novel framework for the recognition of facial expressions at arbitrary poses that is based on 2D geometric features. We address the problem by first mapping the 2D locations of landmark points of facial expressions in non-frontal poses to the corresponding locations in the frontal pose. Then, recognition of the expressions is performed by using any state-of-the-art facial expression recognition method (in our case, multi-class SVM). To learn the mappings that achieve pose normalization, we use a novel Gaussian Process Regression (GPR) model which we name Coupled Gaussian Process Regression (CGPR) model. Instead of learning single GPR model for all target pairs of poses at once, or learning one GPR model per target pair of poses independently of other pairs of poses, we propose CGPR model, which also models the couplings between the GPR models learned independently per target pairs of poses. To the best of our knowledge, the proposed method is the first one satisfying all: (i) being face-shape-model-free, (ii) handling expressive faces in the range from -45° to $+45^\circ$ pan rotation and from -30° to $+30^\circ$ tilt rotation, and (iii) performing accurately for continuous head pose despite the fact that the training was conducted only on a set of discrete poses.

1 Introduction

Facial expression recognition has attracted significant attention because of its usefulness in many applications such as human-computer interaction, face animation and analysis of social interaction [1,2]. Most existing methods deal with images (or image sequences) in which depicted persons are relatively still and exhibit posed expressions in nearly frontal view [1]. However, most of real-world applications relate to spontaneous human-to-human interactions (e.g., meeting summarization, political debates analysis, etc.), in which the assumption of having immovable subjects is unrealistic. This calls for a joint analysis of head pose and facial expressions. Nonetheless, this remains a significant research challenge mainly due to the large variation in the appearance of facial expressions in different views and the difficulty in decoupling these different sources of variation.

Most of the existing approaches that perform pose-invariant facial expression recognition are based on 3D face models. For example, Chang et al. [3] built

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 350–363, 2010.

a probabilistic model on the generalized expression manifold obtained from 3D facial expression range data to recognize the prototypic facial expressions. To the same aim and to analyze the dynamics of facial expressions, Sun and Yin [4] applied 3D dynamic facial surface descriptors. Furthermore, several works proposed to apply 3D Active Appearance Models (AAM) for pose-invariant facial expression analysis (e.g. Sung and Kim [5], Cheon and Kim [6]). Zhu and Ji [7] used 3D Point Distribution Model (3D-PDM) and normalized SVD to recover the facial expression and pose. Wang and Lien [8] used similar 3D-PDM to separate the rigid head rotation from non-rigid facial expressions. Kumano et al. [9] applied a rigid face shape model to build person-dependent descriptors that were later used to decompose facial pose and expression simultaneously. Despite the fact that 3D face models have advantage over 2D approaches in that the effect of head pose on the facial expression analysis can be removed (although this usually comes at the expense of the recovered facial expression accuracy), the main disadvantage is the use of generative models and fitting techniques that can fail to converge. Also, most of these methods are computationally expensive and in need of time-consuming initialization process (e.g. due to manual annotation of more than 60 facial landmark points). Moreover, some of the aforementioned methods such as AAM need to be trained for each person/ facial expression/ head pose separately which makes those methods difficult to apply in real-world applications where unknown subjects/ expressions can be expected.

In contrast to increasing interest in pose-invariant facial expression analysis based on 3D and 2D face-shape models, pose-invariant facial expression analysis based on 2D shape-free methods has been scarcely investigated. This is mostly due to the fact that rigid head motions and non-rigid facial expressions are non-linearly coupled in 2D and difficult to decouple using existing algorithms [7]. For this reason, most of the proposed 2D pose-invariant methods address the problem of (expressionless) face recognition but not the problem of facial expression recognition (e.g. [10]). To the best of our knowledge, the only work that analyzed the problem of pose-invariant facial expression recognition using a 2D shape-free approach is the work by Hu et al. [11]. They proposed a set of pose-wise facial expression classifiers that are used to discriminate simultaneously facial expressions and horizontal head orientations at five pan angles (0°, 30°, 45°, and 90°). However, the performance of this method has not been analyzed for unknown head poses, i.e. poses that were not used to train the classifiers. Moreover, because the classifiers were trained pose-wise, it is not possible to perform recognition of facial expressions that were not included in the training dataset for the given pose (in other words, this facial expression recognition method cannot generalize across poses).

In this paper we propose a 2D face-shape-free method for pose-invariant facial expression recognition. We address the problem by mapping 2D facial points (e.g., mouth corners) from non-frontal poses to the frontal pose where the recognition of facial expressions can be performed by using any state-of-the art facial expression recognition method. The proposed three-step approach is illustrated in Fig. 1. In the first step, we perform head pose estimation by projecting the

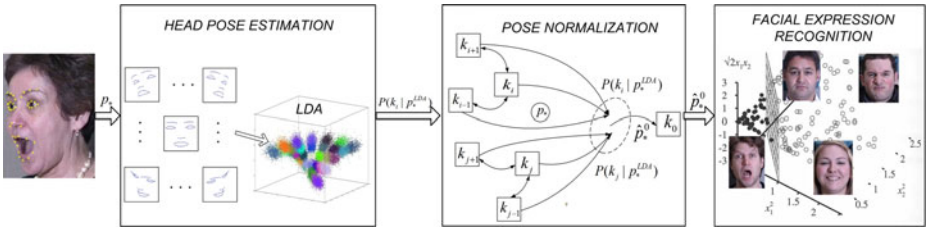


Fig. 1. The overview of the proposed three-step approach. Legend: p_* are the 2D locations of facial landmarks from the input facial image, $P(k_i|p_*^{LDA})$ is the probability of p_* belonging to the pose k_i, where k_0 is the frontal pose. The bidirectional lines in the pose normalization step represent the coupled head poses, and the directed lines represent the CGPR models learned per pair of poses (k_i, k_0). $\hat{p}_*^0$ is the prediction of p_* in the frontal pose obtained as a linearly weighted combination of the aforementioned CGPR models where the weights are proportional to $P(k_i|p_*^{LDA})$.

input datum (i.e. 2D facial points locations) to a low-dimensional manifold (attained by the means of multi-class LDA) and by estimating the probability of it belonging to each of the discrete poses for which training data are available. In the second step, we use the novel Coupled Gaussian Process Regression (CGPR) model to perform pose normalization, that is, to learn mappings between the 2D locations of landmark points of the facial expressions in non-frontal poses and their locations in the frontal pose. Instead of using single Gaussian Process Regression (GPR) model for all target pairs of poses at once, or using only one GPR model per target pair of poses, we propose CGPR models, which also model the couplings between the GPR models learned independently per target pairs of poses. To enable accurate performance for continuous head pose (i.e. for unknown poses), the predictions of the facial landmark locations in the frontal pose obtained by CGPR models from different poses are linearly combined (where the weights are based on head-pose probabilities obtained by the pose estimator in the first step of the proposed approach). The last step in our approach is facial expression classification in the frontal pose attained using the multi-class Support Vector Machine classifier.

The contributions of the proposed methodology are summarized as follows.

1. We propose a 2D face-shape-model-free approach to pose-invariant facial expression recognition that can handle expressive faces in the range from $-45°$ to $+45°$ pan rotation and from $-30°$ to $+30°$ tilt rotation. The proposed approach performs accurately for continuous head pose despite the fact that the training was conducted only on a set of discrete poses. It can also handle successfully the problem of having an unbalanced training dataset (i.e., when examples of certain facial expression categories are not included in the training dataset for a given discrete pose).
2. We propose a novel head pose normalization approach based on the linearly weighted combination of the newly proposed Coupled Gaussian Process Regression (CGPR) models, which model the couplings between the GPR

models learned per target pairs of poses. We employ GPR model since it provides not only the predictions of the facial landmark points in the frontal pose but also the uncertainty in these predictions (obtained through its covariance function) [12]. Moreover, the couplings between the GPR models can be embedded in their covariance structure in a very natural and straightforward manner. Although CGPR is a multiple-output GPR model, it does not model the dependences between its outputs (as done by the dependent-output GPR model such as the one proposed in [13]). Instead, CGPR models the dependences between the predictions obtained by different GPR models (i.e., GPR models learned for different poses). For these newly proposed CGPR models, we show experimentally that the proposed scheme outperforms a linearly weighted combination of GPR models learned per target pairs of poses which, in turn, outperforms baseline methods for pose normalization as 2D- and 3D-PDM.

The rest of the paper is organized as follows. In Section 2 we present our approach to pose-invariant facial expression recognition. In Section 3 we describe the newly proposed CGPR model. Experimental studies are discussed in Section 4, while Section 5 concludes the paper.

2 Pose-Invariant Facial Expression Recognition

In this section we describe a novel 2D face-shape-model-free approach to pose-invariant facial expression recognition given the 2D locations $p \in \mathbf{R}^d$ of $L = d/2$ facial landmarks of a face at an arbitrary pose. The proposed approach consists of three main steps: (i) head pose estimation by using a pose classifier on p, (ii) pose normalization by mapping the positions p of the facial landmarks from a non-frontal pose to the corresponding 2D positions p^0 in the frontal pose, and (iii) facial expression classification in the frontal pose. These steps are described in detail in the following sections and are summarized in Alg. 1. The theory behind the second step, that is the proposed CGPR model, is described in detail in Section 3.

In what follows, we assume that we have training data for each of P discrete poses and the correspondences between the points for each target pair of poses (non-frontal and frontal pose). In our case, we discretized the head pose space which resulted in $P = 35$ poses evenly distributed across the range from $-45°$ to $+45°$ pan rotation and from $-30°$ to $+30°$ tilt rotation. We denote by $D^k = \{p_1^k, ..., p_{N_k}^k\}$ the data set from pose k, and by $D = \left\{D^0, ..., D^k, ..., D^{P-1}\right\}$ the whole training data set, where N_k represents the number of training data in the pose k.

2.1 Head Pose Estimation

Various head pose estimation methods based on appearance and/or geometric features are proposed in the literature [14]. We propose to estimate the probability of each head pose belonging to a discretized head-pose space represented by

a low-dimensional manifold attained by means of multi-class LDA. Firstly, we normalize all examples from D (2D locations of facial landmarks in P poses), to remove the scale and translation components, as described in [15]. Secondly, to learn the manifold from such normalized data, we employ multi-class LDA since it is a simple linear transform that, given a training set with known pose labels, finds a low dimensional manifold which best represents pose variations while ignoring variations due to facial expressions. The estimated probability of input 2D facial points locations p being in pose k is given by $P(p^{lda}|k) = G(p^{lda}; \mu^k, \Sigma^k)$, where G is a normal density centered at μ^k and with covariance Σ^k. p^{lda} is the projection of p onto the low dimensional manifold. By applying Bayes' rule, we obtain the probability of being in pose k as $P(k|p^{lda}) \propto P(p^{lda}|k)P(k)$, where we assume a uniform prior $P(k) = 1/P$.

2.2 Head Pose Normalization

Given input data p_* containing the 2D locations of the facial points in an unknown head pose, our goal is to predict the location of these points in the frontal pose $\hat{p}_*^0$. To this end, we learn the functions $f_C^{(k)}(p_*)$ $(1 \leq k \leq P)$ which are later used to make predictions for input data p_*. These functions are modeled by the proposed CGPR models described in detail in Section 3. Thus, given p_*, $P(k|p_*^{lda})$ and $f_C^{(k)}(p_*)$, we obtain the locations of the frontal facial landmarks $\hat{p}_*^0$ as a linearly weighted combination of $f_C^{(k)}(p_*)$ for all k which satisfy $P(k|p_*^{lda}) > P_{min}$, where the weights are proportional to the head pose probabilities $P(k|p_*^{lda})$. The mathematical formulation of this is given in Step 2 in Alg. 1. Let us mention here that before $f_C^{(k)}$ is applied to p_*, it is registered to a reference face in pose k using a simple affine transform. The latter is calculated using five referential points: the nasal spine point and the inner and outer corners of the eyes (because they are stable facial points and the contractions of the facial muscles do not affect them).

2.3 Facial Expression Classification in Frontal Pose

We address the problem of pose-invariant facial expression recognition by performing pose normalization first, and subsequently applying any 2D-geometric-feature-based facial expression recognition method to the normalized input data (see [1]). In this paper, we use the multi-class SVM with decision function is given by

$$l = \arg\max_z \Big(\sum_{i:p_i^0 \in T_z} \alpha_i K(p_i^0, \hat{p}_*^0) + b_z \Big), \quad z = 1...Z, \tag{1}$$

where α_i and b_z are the weight and bias parameters, and $K(p_i^0, \hat{p}_*^0)$ is a vector of inner products between the training data $p_i^0 \in D^0$, containing Z facial expressions, and an estimate of p_* in the frontal pose, $\hat{p}_*^0$. The set T_z contains data points that depict facial expression z.

Algorithm 1. Pose-Invariant Facial Expression Recognition

Input: Positions of facial landmarks in an unknown pose (p_*)
Output: Facial expression label (l)
1. Apply the pose estimation (Sec. 2.1) to obtain $P(k|p_*^{lda})$, $k = 0\,..\,P-1$
2. Register p_* to poses $k \in \mathcal{K}$ which satisfy $P(k|p_*^{lda}) > P_{min}$ (Sec. 2.2), and predict the locations of the facial landmarks points in frontal pose

$$\hat{p}_*^0 = \frac{1}{\sum_{k\in\mathcal{K}} P(k|p_*^{lda})} \sum_{k\in\mathcal{K}} P(k|p_*^{lda}) f_C^{(k)}(p_*^k)$$

3. Facial expression classification in frontal pose (Sec. 2.3)

$$l \leftarrow \arg\max_z \Big(\sum_{i:p_i^0 \in T_z} \alpha_i K(p_i^0, \hat{p}_*^0) + b_z \Big)$$

3 Coupled Gaussian Process Regression (CGPR)

In this section we describe a novel methodology for learning functions that map the 2D locations of facial points p in non-frontal poses to the corresponding 2D locations in the frontal pose. We learn a set of such functions, denoted by $f_C^{(k)}$, each one of which is associated with a certain pose k, where k is one of the discrete poses P for which training examples are available (i.e. $0 \leq k \leq P-1$). Roughly speaking, $f_C^{(k)}(p_*)$ is expected to provide good mappings for p_* obtained at an arbitrary pose that is relatively close to the pose k.

In order to learn $f_C^{(k)}$, we learn a set of $P-1$ mapping functions $\{f^{(1)}, .., f^{(P-1)}\}$ first, where the function $f^{(k)}$ maps the positions of the landmark points p^k in pose k to the corresponding points p^0 in the frontal pose. $f^{(k)}$ is learned using a GPR model for the target pair of poses $(k, 0)$ based on the datasets D^k and D^0, i.e., the sets that contain landmark points p in pose k and in the frontal pose denoted by 0.

3.1 Gaussian Process Regression (GPR)

In this section we describe the base GPR model for learning the mapping functions f^k. Formally, given a set of N_k examples of facial images containing the landmark locations in pose k, and the corresponding landmark locations in the frontal pose 0 (i.e. $\{D^k, D^0\}$), we learn the function $f^{(k)}$: $\mathbb{R}^d \rightarrow \mathbb{R}^d$ that maps $p_i^k \in D^k$ to $p_i^0 \in D^0$, where $i = 1..N_k$. Assuming Gaussian noise ε_i with zero mean and covariance matrix $\sigma_n^2 I$, this is expressed by $p_i^0 = f^{(k)}(p_i^k) + \varepsilon_i$. In GPR model, a zero mean Gaussian process prior is placed over the function $f^{(k)}$, that is $f^{(k)} \sim GP(0, K + \sigma_n^2 I)$, where $K(D^k, D^k)$ denotes $N_k \times N_k$ matrix of the covariances evaluated at pairs (p_i^k, p_j^k) by applying the kernel

$$k(p_i^k, p_j^k) = \sigma_s^2 \exp(-\frac{1}{2}(p_i^k - p_j^k)^T W (p_i^k - p_j^k)) + \sigma_l p_i^k p_j^k + \sigma_b, \qquad (2)$$

where $i, j = 1..N_k$. σ_s and $W = diag(w_1, ..., w_d)$ are the parameters of the radial basis function with different length scales for each input dimension (each coordinate of each landmark point), σ_l is the process variance which controls

the scale of the output function $f^{(k)}$, and σ_b is the model bias. This kernel has been widely used due to its ability to handle both linear and non-linear data structures [16]. During inference, we obtain the predictive mean $f^{(k)}(p_*^k)$ and the corresponding variance $V^{(k)}(p_*^k)$ for a new input p_*^k as

$$f^{(k)}(p_*^k) = k_*^T(K + \sigma_n^2 I)^{-1} D^0 \tag{3}$$

$$V^{(k)}(p_*^k) = k(p_*^k, p_*^k) - k_*^T(K + \sigma_n^2 I)^{-1} k_* \tag{4}$$

where $k_* = k(D^k, p_*^k)$, and $k(\cdot,\cdot)$ is given by Eq.(2). The kernel parameters $\theta = \{\sigma_s, W, \sigma_l, \sigma_b, \sigma_n\}$ are found by maximizing the log marginal likelihood of the training outputs using the conjugate gradient algorithm [12]. We assume here that the output dimensions (each coordinate of each landmark point in p_i^0) are *a priori* identically distributed [12]. This allows us to easily handle multiple outputs by applying the same covariance matrix to each output.

3.2 Learning Couplings

The mapping functions $\{f^{(1)}, ..., f^{(k)}, ..., f^{(P-1)}\}$ are learned independently for each target pair of poses; however, they need not be independent. Moreover, if the outputs obtained by different mapping functions are correlated, inferring the couplings between them may help obtain better predictions [17]. We model the coupling between two functions, $f^{(k_1)}$ and $f^{(k_2)}$, for pose k_1, using Gaussian distribution on the differences of their predictions obtained by evaluating these functions on the training data D^{k_1}. It is expressed by

$$P(f^{(k_1)}, f^{(k_2)}|k_1) \propto \exp(-\frac{1}{2} d^T \Sigma^{-1} d), \tag{5}$$

where $d = f^{(k_1)}(p_*^{k_1}) - f^{(k_2)}(p_*^{k_1})$, and $\Sigma = \sigma^2_{(k_1,k_2)} I$. The variance $\sigma^2_{(k_1,k_2)}$ measures the extent to which $f^{(k_2)}$ is coupled (i.e., similar) to $f^{(k_1)}$. Alternatively, this can be seen as an independent noise component in the predictions made by $f^{(k2)}$ because it is evaluated on data from different pose, i.e., pose k_1. Since we assume that this noise is Gaussian and independent of the noise process modeled by $f^{(k_2)}$, their covariances can simply be added [12]. Accordingly, we update the mean and variance given by Eq.(3) and Eq.(4), respectively, to obtain the mean and variance of CGPR model, that is

$$f^{(k_1,k_2)}(p_*^{k_1}) = k_*^T(K_2 + (\sigma_n^2 + \sigma^2_{(k_1,k_2)})I)^{-1} D^0 \tag{6}$$

$$V^{(k_1,k_2)}(p_*^{k_1}) = k(p_*^{k_1}, p_*^{k_1}) - k_*^T(K_2 + (\sigma_n^2 + \sigma^2_{(k_1,k_2)})I)^{-1} k_*, \tag{7}$$

where $k_* = k(D^{k_2}, p_*^{k_1})$. It is clear that the smaller the coupling between the functions $f^{(k_1)}$ and $f^{(k_2)}$, the higher the uncertainty in the predictions obtained by $f^{(k_1,k_2)}$. In the case of perfect coupling (when $\sigma^2_{(k_1,k_2)} \to 0$), we do not increase the uncertainty in the predictions obtained by $f^{(k_1,k_2)}$ (which converges to $f^{(k_2)}$).

On the other hand, when there is no coupling ($\sigma^2_{(k_1,k_2)} \to \infty$), we obtain the prior mean and covariance of $f^{(k_2)}$ ($f^{(k_1,k_2)}(p_*^{k_1}) \to 0$ and $V^{(k_1,k_2)}(p_*^{k_1}) \to k(p_*^{k_1}, p_*^{k_1})$). Because the variance of such prediction is the highest possible (learned by the model), this prediction will be suppressed by the covariance intersection rule described in Sec 3.4. Finally, the covariance matrix of CGPR model is guaranteed to be positive definite (the covariance matrix of the base GPR models learned in Sec. 3.1 is positive definite) since we only add a positive term to the diagonal of the covariance matrix in Eq.(6)&(7) [12].

3.3 Pruning CGPR Models

The couplings between all functions pairs $(f^{(k_1)}, f^{(k_2)})$ can be easily learned. Nevertheless, inference utilizing all them would be slow. Also, not all of the coupled functions $f^{(k_1,k_2)}$ contribute significantly in reducing the uncertainty in the predictions. As a pruning criterion, we propose using a measure based on the number of effective degrees of freedom of a GP [18]. In the framework of CGPR model, it has the following form

$$C_{eff}^{(k_1,k_2)} = \sum_{i=1}^{N_{k_2}} \frac{\lambda_i}{\lambda_i + \sigma_n^2 + \sigma^2_{(k_1,k_2)}} \tag{8}$$

where λ_i are the eigenvalues of the matrix K_2. If $\sigma^2_{(k_1,k_2)}$ is large, $C_{eff}^{(k_1,k_2)} \to 0$ and the predictions made by $f^{(k_1,k_2)}$ can be neglected. We compare the ratio $C_{eff}^{(k_1,k_2)}/C_{eff}^{(k_1)}$ to a threshold C_{min} to decide which coupled functions are relevant.

3.4 Covariance Intersection (CI)

In this section we describe how to fuse the predictions obtained by different mapping functions $f^{(k_1)}$ and $f^{(k_1,k_2)}$ in order to obtain a single prediction $f_C^{(k_1)}$ associated with pose k_1. A straightforward solution would be to select weighting functions inversely proportional to the variance of the predictions obtained by the individual functions. However, this fusion rule is optimal only if the predictions (i.e. their errors) are uncorrelated [17]. Since for a query point p_* we do not *a priori* know whether the predictions are correlated or not, the above fusion rule may not be optimal. Recently, a fusion rule, called Covariance Intersection (CI), for combining predictions in the presence of unknown cross covariance, has been proposed in [19]. To illustrate this, consider two GPR models, $f^{(k_1)}$ and $f^{(k_1,k_2)}$, with the mean and covariance pairs, $\{f^{(k_1)}(p_*), V^{(k_1)}(p_*)\}$ and $\{f^{(k_1,k_2)}(p_*), V^{(k_1,k_2)}(p_*)\}$. The CI yields the mean and covariance pair $\{f_C^{(k_1)}(p_*), V_C^{(k_1)}(p_*)\}$ obtained as

$$V_C^{(k_1)^{-1}}(p_*) = \omega(V^{(k_1)}(p_*))^{-1} + (1-\omega)(V^{(k_1,k_2)}(p_*))^{-1} \tag{9}$$

$$f_C^{(k_1)}(p_*) = V_C^{k_1}(p_*)(\omega(V^{(k_1)}(p_*))^{-1} f^{(k_1)}(p_*) + (1-\omega)(V^{(k_1,k_2)}(p_*))^{-1} f^{(k_1,k_2)}(p_*)) \tag{10}$$

where $\omega \in [0, 1]$ is a scalar that minimizes some criterion of uncertainty. In all our experiments we minimize the trace of $V_C^{k_1}(p_*)$ that we use as the uncertainty criterion, as proposed in [19].

4 Experiments

The experimental evaluation of the proposed methodology has been carried out using two datasets: the BU-3D Facial Expression (BU3DFE) database [20] containing 3D range data and the CMU Pose, Illumination and Expression Database (MultiPie) [21] containing multi-view facial expression data. BU3DFE contains 3D scans of 7 facial expressions, Angry, Disgust, Fear, Happy, Sad, Surprise and Neutral, performed by 100 subjects (60% female of various ethnic origin). All facial expressions except Neutral were sampled in four different levels of intensity. We generate 2D multi-view images of facial expressions from the available 3D data by rotating 39 facial landmark points provided by the database creators (see Fig. 3), which were used further as the features in our study. The data in our experiments include images of 50 subjects (54% female) at $\pm 15°$,$\pm 30°$ and $\pm 45°$ pan angles, and $\pm 15°$ and $\pm 30°$ tilt angles (see Fig. 1), with 5° increment, resulting in 1250 images for each of 247 poses. The training data are subsampled from this dataset to include images of expressive faces in 35 poses (15° increment in pan and tilt angles). These data (referred to as BU-TR dataset in the text below) as well as the rest of the data (referred to as BU-TST dataset and used to test the performance of the proposed methods) were partitioned into five folds in a person-independent manner for use in a 5-fold cross validation procedure. To evaluate the performance of the method in case of real data (as opposed to synthetic BU-TR/TST data), we used a subset of MultiPie containing images of 50 subjects (22% female) displaying 4 expressions (neutral, disgust, surprise, and happy) captured at 4 pan angles (0°,$-15°$,$-30°$ and $-45°$), resulting in 200 images per pose. All images were hand labeled in terms of 39 landmark points and the dataset was partitioned in a person-independent manner for use in a 5-fold cross validation procedure.

The rest of this section is organized as follows. First we present the experiments aimed at evaluation of the accuracy of the proposed head pose normalization method. To measure the accuracy of the method, we used the root-mean squared (RMSE) distance between the predicted image positions of the facial landmarks in the frontal pose and the ground truth (the manually annotated facial landmarks in frontal pose). As suggested by the results attained when testing on BU-TST dataset (see Fig. 2), the proposed CGPR-based method outperforms both GPR-based method and the 'baseline' methods for pose normalization, namely, 2D-PDM [22] and 3D-PDM [7]. The superior performance of the proposed CGPR-based method is also shown in the case of noisy data (see Table 1). Secondly, we evaluate the performance of the proposed pose-invariant facial expression recognition method. Testing was performed on faces from BU-TST images in (i) frontal pose (FP), (ii) non-frontal training poses (tp), and (iii) unknown poses (ntp), where the pose normalization was achieved using

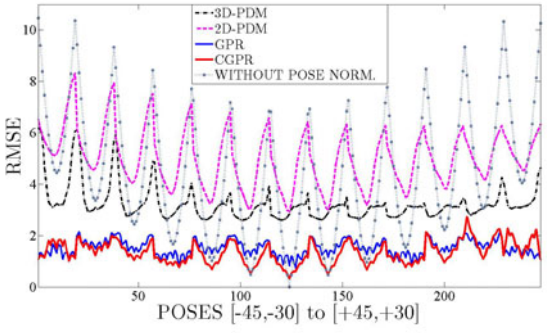


Fig. 2. Comparison of head pose normalization methods CGPR, GPR, 3D-PDM and 2D-PDM, trained on BU-TR (35 head poses) and tested on BU-TST (247 head poses) in a person-independent manner in terms of RMSE

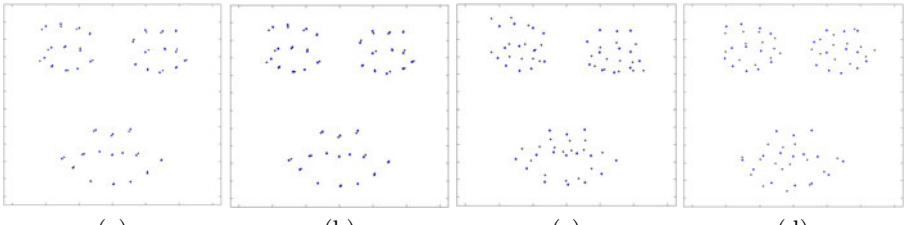


Fig. 3. Prediction of the facial landmarks in the frontal pose for an BU3DFE image of Happy facial expression in pose $(-45°, -30°)$ obtained by using (a) CGPR (b) GPR (c) 3D-PDM and (d) 2D-PDM. The blue $\bigtriangledown$ represent the ground truth and the black $\square$ are the predicted points. As can be seen, the alignment of the predicted and the corresponding ground truth facial landmarks is far from perfect in case of 3D/2D-PDM.

the CGPR-based method (Table 2). Finally, to evaluate the performance of the method in case of real data and in case of unbalanced data (i.e. when the method is trained on data where some of facial expression categories are missing in certain poses), we carry out experiments on MultiPie dataset (Table 3). For all experiments carried out on BU-TR/TST datasets, we did the following: the head pose estimator was trained on BU-TR dataset and when tested on BU-TST data, it predicted the correct (closest) head pose in 95% of cases. The base GPR models in Alg. 2 were trained on BU-TR dataset for each of the 34 target pairs of poses. Furthermore, we set P_{min} in Alg. 1 and C_{min} in Alg. 2 to experimentally found optimal values that are 0.1 and 0.75, respectively. The 2D-PDM and 3D-PDM were trained using the frontal data from BU-TR dataset (for 3D-PDM, the corresponding 3D data were used), retaining 15 modes of non-rigid shape variation.

Table 1. Comparison of head pose normalization methods CGPR, GPR, 3D-PDM and 2D-PDM, trained on BU-TR (35 head poses) and tested on BU-TST (247 head poses) corrupted with different levels of Gaussian noise with standard deviation σ, in a person-independent manner, in terms of RMSE

	$\sigma = 0$		$\sigma = 0.5$		$\sigma = 1$		$\sigma = 2$		$\sigma = 3$	
	tp	ntp	tp	ntp	tp	ntp	tp	ntp	tp	ntp
3D-PDM	3.1±0.9	3.2±0.6	3.1±0.8	2.9±0.6	3.2±0.7	2.9±0.6	3.7±0.9	3.3±0.6	4.0±0.5	3.8±0.5
2D-PDM	5.2±1.2	4.9±1.1	5.4±1.4	5.3±1.3	5.7±1.4	5.4±1.3	6.0±1.3	5.8±1.3	6.3±1.2	6.1±1.1
GPR	1.1±0.2	1.6±0.3	1.3±0.3	1.5±0.3	1.6±0.2	1.8±0.3	2.4±0.2	2.5±0.2	3.3±0.1	3.4±0.1
CGPR	1.1±0.3	1.4±0.4	1.2±0.3	1.4±0.2	1.5±0.2	1.6±0.2	2.3±0.2	2.4±0.2	3.2±0.2	3.3±0.1

Table 2. Facial expression recognition results using 7-class-SVM trained on frontal-pose expressive images from BU-TR and tested on BU-TST images in (i) frontal pose (FP), (ii) non-frontal training poses (tp), and (iii) unknown poses (ntp), where the pose normalization was achieved using the CGPR-based method. The best results reported by Hu *et al.* [11] for BU3DFE are reported for comparison purposes. All results are given in terms of correct recognition rate percentages.

	Disgust	Angry	Fear	Happy	Sad	Surprise	Neutral
FP+SVM	74.5±2.1	69.9±1.8	58.3±1.2	80.4±2.1	76.3±2.0	91.1±1.4	73±2.5
CGPR+SVM (tp)	71.0±3.1	72.8±1.6	58.0±1.7	81.9±2.9	73.8±2.7	89.9±1.9	73±3.0
CGPR+SVM (ntp)	70.1±3.4	71.1±2.2	56.2±2.2	80.2±1.8	72.1±2.9	88.1±2.0	72±2.4
Hu *et al.* [11] (tp)	69.3	71.3	52.5	78.3	71.5	86.0	-

Evaluation of the accuracy of the proposed head pose normalization method – Fig. 2 shows the comparative results in terms of RMSE of the tested head pose normalization methods along with the results obtained when no pose normalization is performed and only the translation component has been removed. As can be seen, both GPR- and CGPR-based methods significantly outperform the 2D/3D 'baseline' methods for pose normalization. Judging from Fig 3, this is probably due to the fact that the tested 2D/3D deformable face-shape-based models were not able to accurately model the non-rigid facial movements present in facial expression data. The performance of the aforementioned models in the presence of noise in test data was evaluated on BU-TST data corrupted by adding four different levels of noise. As can be seen from Table 1, even in the presence of high levels of noise the performance of GPR/CGPR-based methods is comparable to that of 2D/3D-PDM achieved for noise-free data. The performance of GPR- and CGPR-based methods is highly comparable in the aforementioned experiments where the utilized data were balanced (i.e. when the method is trained on data containing examples of all facial expression categories in all target poses). However, the results shown in Table 1 (i.e. when no noise is present in the data) suggest that the proposed CGPR-based method slightly outperforms the GPR-based method when tested on unknown poses (ntp).

Table 3. Facial expression recognition results using 4-class-SVM trained and tested on unbalanced data from MultiPie, where the pose normalization was achieved using GPR- or CGPR-based method. The unbalanced dataset was prepared by removing all examples of one facial expression category from one of the non-frontal poses. The testing was performed on the removed examples in a cross-validation person-independent manner for each expression and each pose. The performance of the classifier trained/tested on frontal-pose expressive images from MultiPie is also reported for the purposes of baseline comparison. All results are given in terms of correct recognition rate percentages.

		Disgust	Happy	Surprise	Neutral
FP+SVM	RR[%]	94.2±2.3	95.6±1.3	97.4±0.9	93.7±1.9
GPR+SVM	RR[%]	68.9±6.2	74.2±4.5	69.4±5.2	73.8±3.9
	RMSE	3.10±0.7	3.21±0.9	3.62±0.9	2.80±0.7
CGPR+SVM	RR[%]	85.2±4.3	90.2±3.1	89.8±3.2	88.2±3.1
	RMSE	1.95±0.3	1.80±0.4	2.40±0.3	1.90±0.4

Algorithm 2. Learning and inference with CGPR

OFFLINE: Learning base GPR models and coupling parameters
1. Learn $P-1$ base GPR models $\{f^{(1)}, ..., f^{(P-1)}\}$ for target pairs of poses (Sec. 3.1)
2. Perform coupling of base GPR models learned in Step 1
   for k_1=1 to P-1 **do**
      for k_2=1 to P-1 & $k_1 \neq k_2$ **do**
         estimate $\sigma_{(k_1,k_2)}$ (Sec. 3.3)
         if $C_{eff}^{(k_1,k_2)} > C_{min}$ **then** $\sigma_C^{k_1} = [\sigma_C^{k_1}, \sigma_{(k_1,k_2)}]$ **end if**
      end for
      store $\sigma_C^{k_1}$
   end for

ONLINE: Inference of the facial landmarks $p_*^{k_1}$ in pose k_1
S_{k_1}: number of the functions coupled to $f^{(k_1)}$
1. Evaluate base GPR model for pose k_1 (Sec. 3.1): $Pr(0) = \{f^{(k_1)}(p_*^{k_1}), V^{(k_1)}(p_*^{k_1})\}$
2. Evaluate CGPR models for pose k_1 (Sec. 3.3)
   for i=1 to S_{k_1} **do** $\sigma_{(k_1,i)} = \sigma_C^{k_1}(i)$, $Pr(i) = \{f^{(k_1,i)}(p_*^{k_1}), V^{(k_1,i)}(p_*^{k_1})\}$ **end for**
3. Combine estimates using CI (Sec. 3.4): $\{f_C^{(k_1)}(p_*^{k_1}), V_C^{(k_1)}(p_*^{k_1})\} = CI(Pr)$

Evaluation of the proposed pose-invariant facial expression recognition method – The results presented in Table 2 clearly suggest that the proposed pose-invariant facial expression recognition method performs accurately for continuous head pose (i.e. for unknown poses; ntp-case in the Table 2) despite the fact that the training was conducted only on a set of discrete poses (i.e. on BU-TR). As can be seen further from Table 2, even in case of unknown poses, the proposed method outperforms the method reported by Hu *et al.* [11], where pose-wise SVM classifiers were trained and tested only on known poses. While the aforementioned experiments suggest that the performance of GPR- and CGPR-based methods is highly comparable when the utilized data are balanced, the same is not the case when the utilized data are unbalanced.

Specifically, the results presented in Table 3 clearly suggest that the proposed CGPR-based pose-invariant facial expression recognition method significantly outperforms the GPR-based method in case of unbalanced data, i.e., when one facial expression category is missing in a certain pose. Judging from Table 3, RMSE rows, the reason for this is that the CGPR-based head pose normalization is significantly better than that obtained by the GPR-based method. In turn, this can be explained by the non-parametric nature of the GPR-based method due to which it cannot generalize well beyond the training data. On the contrary, the CGPR-based method overcomes this by employing the knowledge (training data) provided by the underlying CGPR models.

5 Conclusion

We presented a novel 2D-shape-free method for the recognition of facial expressions at arbitrary poses that is based on pose normalization of 2D geometric features. For pose normalization, we proposed Coupled Gaussian Process Regression (CGPR) model that learns direct mappings between the facial positions at an arbitrary pose and the positions in the frontal pose. Experimental results demonstrate the advantages of the proposed pose normalization in comparison to generative methods and its robustness to incomplete training data (i.e. expressions and poses that do not belong to the training dataset). For the problem of expression recognition, the proposed method is shown to demonstrate classification performance comparable to the ones obtained by pose-specific classification schemes for the significantly more difficult problem of expression recognition at an unknown pose.

Acknowledgments. This work is funded in part by the European Community's 7th Framework Programme [FP7/2007-2013] under grant agreement no. 211486 (SEMAINE), and in part by the European Research Council under the ERC Starting Grant agreement no. ERC-2007-StG-203143 (MAHNOB). The work of Ioannis Patras is partially supported by EPSRC project EP/G033935/1.

References

1. Zeng, Z., Pantic, M., Roisman, G.I., Huang, T.S.: A survey of affect recognition methods: Audio, visual, and spontaneous expressions. IEEE Trans. Pattern Analysis and Machine Intelligence 31, 39–58 (2009)
2. Vinciarelli, A., Pantic, M., Bourlard, H.: Social signal processing: Survey of an emerging domain. Image and Vision Computing 27, 1743–1759 (2009)
3. Chang, Y., Vieira, M., Turk, M., Velho, L.: Automatic 3d facial expression analysis in videos. In: Proc. Int'l Workshop Analysis and Modelling of Faces and Gestures, pp. 293–307 (2005)
4. Sun, Y., Yin, L.: Facial expression recognition based on 3d dynamic range model sequences. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 58–71. Springer, Heidelberg (2008)

5. Sung, J., Kim, D.: Real-time facial expression recognition using staam and layered gda classifier. Image and Vision Computing 27, 1313–1325 (2009)
6. Cheon, Y., Kim, D.: Natural facial expression recognition using differential-aam and manifold learning. Pattern Recognition 42, 1340–1350 (2009)
7. Zhu, Z., Ji, Q.: Robust real-time face pose and facial expression recovery. In: Proc. Int'l Conf. Computer Vision and Pattern Recognition, vol. 1, pp. 681–688 (2006)
8. Wang, T.H., Lien, J.J.J.: Facial expression recognition system based on rigid and non-rigid motion separation and 3d pose estimation. Pattern Recognition 42, 962–977 (2009)
9. Kumano, S., Otsuka, K., Yamato, J., Maeda, E., Sato, Y.: Pose-invariant facial expression recognition using variable-intensity templates. Int'l J. Computer Vision 83, 178–194 (2009)
10. Chai, X., Shan, S., Chen, X., Gao, W.: Locally linear regression for pose-invariant face recognition. IEEE Trans. Image Processing 16, 1716–1725 (2007)
11. Hu, Y., Zeng, Z., Yin, L., Wei, X., Tu, J., Huang, T.: A study of non-frontal-view facial expressions recognition. In: Proc. Int'l Conf. Pattern Recognition, pp. 1–4 (2008)
12. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press, Cambridge (2005)
13. Boyle, P., Frean, M.: Dependent gaussian processes. In: Advances in Neural Information Processing Systems, vol. 17, pp. 217–224. MIT Press, Cambridge (2005)
14. Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation in computer vision: A survey. IEEE Trans. Pattern Analysis and Machine Intelligence 31, 607–626 (2009)
15. Rudovic, O., Patras, I., Pantic, M.: Facial expression invariant head pose normalization using gaussian process regression. In: Proceedings of IEEE Int'l Conf. Computer Vision and Pattern Recognition, vol. 3 (in Press, 2010)
16. Chen, T., Morris, J., Martin, E.: Gaussian process regression for multivariate spectroscopic calibration. Chemometrics and Intelligent Laboratory Systems 87, 59–71 (2007)
17. Tresp, V., Taniguchi, M.: Combining estimators using non-constant weighting functions. In: Advances in Neural Information Processing Systems, pp. 419–426 (1995)
18. Tresp, V.: A bayesian committee machine. Neural Computing 12, 2719–2741 (2000)
19. Julier, S.J., Uhlmann, J.K.: A non-divergent estimation algorithm in the presence of unknown correlations. In: Proc. American Control Conf., pp. 2369–2373 (1997)
20. Wang, J., Yin, L., Wei, X., Sun, Y.: 3d facial expression recognition based on primitive surface feature distribution. In: Proc. Int'l Conf. Computer Vision and Pattern Recognition, vol. 2, pp. 1399–1406 (2006)
21. Gross, R., Matthews, I., Cohn, J., Kanade, T., Baker, S.: Multi-pie. Image and Vision Computing 28, 807–813 (2010)
22. Cootes, T., Taylor, C.: Active shape models - smart snakes. In: Proc. British Machine Vision Conf., pp. 266–275 (1992)

Bilinear Kernel Reduced Rank Regression for Facial Expression Synthesis

Dong Huang and Fernando De la Torre

Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, USA

Abstract. In the last few years, Facial Expression Synthesis (FES) has been a flourishing area of research driven by applications in character animation, computer games, and human computer interaction. This paper proposes a photo-realistic FES method based on Bilinear Kernel Reduced Rank Regression (BKRRR). BKRRR learns a high-dimensional mapping between the appearance of a neutral face and a variety of expressions (e.g. smile, surprise, squint). There are two main contributions in this paper: (1) Propose BKRRR for FES. Several algorithms for learning the parameters of BKRRR are evaluated. (2) Propose a new method to preserve subtle person-specific facial characteristics (e.g. wrinkles, pimples). Experimental results on the CMU Multi-PIE database and pictures taken with a regular camera show the effectiveness of our approach.

1 Introduction

Photorealistic facial expression synthesis (FES) has recently become an active research topic in computer vision and graphics. Applications of FES can be found in diverse fields such as character animation for movies and advertising, computer games, interactive education [1], video teleconferencing [2], avatars [3,4], and facial surgery planning [5]. Generating photo-realistic facial expressions still remains an open research problem due the uncanny ability of people to perceive subtle details in people's faces.

Learning-based methods (e.g. [6,7]) have become a popular approach for FES. However, the use of these methods has several challenges: (1) Muscle deformations due to expression changes can have a large number of degrees of freedom. There are more than 20 groups of facial muscles innervated by facial nerves [8]. The combinations of their movements are nearly innumerable. To model all this variability learning-based methods typically require large amounts of training samples for accurate FES. (2) Synthesis of some facial expressions requires to model subtle facial deformations, for instance wrinkles during squinting. (3) A good model should be able to decouple the identity of the subject from the expression, pose, and illumination while preserving person-specific details (e.g. pimples, beard). (4) Typically the dimensionality of the images is large in comparison with the amount of training samples which causes over-fitting of the model. To address these problems, this paper proposes Bilinear Kernel Reduced Rank Regression (BKRRR) to learn a nonlinear mapping between the frontal neutral image and images with different facial expressions of a subject. Fig. 1 illustrates the process for FES using BKRRR.

The two main contributions of this paper are: (1) Propose BKRRR for FES. BKRRR learns a nonlinear mapping from a neutral face to other facial expressions (e.g. smile,

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 364–377, 2010.

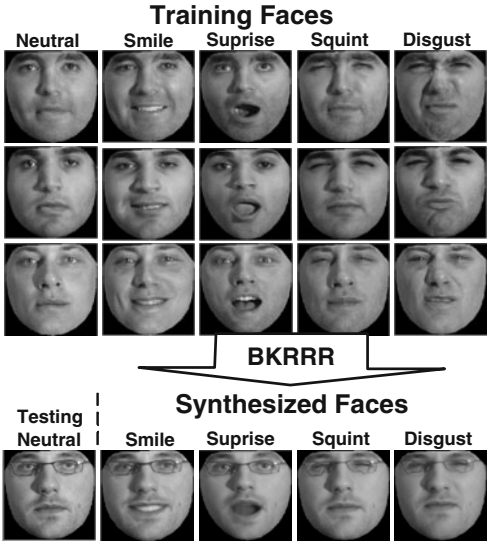


Fig. 1. Synthesizing facial expressions from a neutral face using BKRRR

surprise, squint) that effectively decouples the identity and expression changes. We explore the use of three algorithms for learning the parameters in KRRR and BKRRR, that are based on Subspace Iteration (SI), generalized eigen-decomposition, and Alternated Least Square (ALS). We evaluate the accuracy and computational complexity of each method. (2) Propose a modification of BKRRR to capture subtle person-specific facial features (e.g. glasses, pimples, wrinkles, beard).

The rest of the paper is organized as follows. Section 2 reviews related work on FES. Section 3 describes the KRRR model and three algorithms to learn the KRRR parameters. Section 4 formulates the Bilinear KRRR model, and explores its use to preserve subtle facial details not present in the training samples. Section 5 describes the experimental results, and Section 6 finalizes the paper with the conclusions.

2 Previous Work

Liu et al. [6] proposed a geometric warping algorithm in conjunction with the Expression Ratio Image (ratio between the neutral image and the image of a given expression) to synthesize new expressions preserving subtle details such as wrinkles and cast shadows. Zhang et al. [7] synthesized facial expressions using a local face model. Each region of the face was reconstructed as a convex combination of the corresponding regions in the training set. The synthesized face regions were later blended along the region boundaries. Regression-based approaches find solutions as the weighted combinations of the training data. However, it is unclear how the combination of training data can reproduce subtle local appearance features presented only in the testing samples such as wrinkles, glasses, beard, or pimples. In related work, Nguyen et al. [9] used

extensions of Principal Component Analysis (PCA) to remove glasses and beards in images, and used regression techniques to fill out the missing information.

Tensor-based approaches [10,11,12] perform Higher-Order Singular Value Decomposition (HOSVD) to factorize the normalized face appearance into identity, expression, pose, and illumination. Given the factorization, FES [13,14,15] is done by first computing the identity coefficients of the new testing person, and then reassembling the identity factor with expression factors learned by the HOSVD. A drawback of tensor-based approaches is the need of fully labeled examples across illumination, expressions, and pose. Moreover, it's also unclear how tensor-based methods can preserve subtle person-specific features (e.g. wrinkles, pimples).

Other methods learn the dynamics of the facial expression changes given several video sequences of different subjects performing the same expression. Bettinger et al. [16] used a sampled mean shift and a variable length Markov model to generate person-specific sequences of facial expressions. Zalewski et al. [17] clustered the shape and texture components with a mixture of probabilistic PCA. Each cluster corresponds to a facial expression and clusters are used for FES. Chang et al. [18] introduced a probabilistic model to learn a nonlinear dynamical model on a manifold of expressions containing the neutral and six universal expressions. In the field of computer graphics, several works used 3D models to dynamically animate avatars [19,20,21]. See [22] for a more extensive review of facial expression synthesis methods.

3 Kernel Reduced Rank Regression (KRRR)

Since its introduction in the early 1950s by Anderson [23], the reduced-rank regression (RRR) model has inspired a wealth of diverse applications in several fields such as signal processing [24] (also known as reduced-rank Wiener filtering), neural networks [25] (also known as asymmetric PCA), time series [23], and computer vision [26]. This section describes KRRR and explores three methods to compute its parameters.

3.1 Error Function for Kernel Reduced Rank Regression (KRRR)

Let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \Re^{d_x \times n}$ (see the footnote for notation[1]) be a matrix containing the vectorized images of neutral faces for n subjects, and $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_n] \in \Re^{d_y \times n}$ contains the vectorized images of the same subjects with a different expression.

Due to lack of training samples to constrain the regression parameters, learning a linear regression between two high-dimensional data sets is usually an ill-posed problem. Consider learning the regression matrix $\mathbf{T}$ that optimizes $\min_{\mathbf{T}} ||\mathbf{Y} - \mathbf{T}\mathbf{X}||_F^2$. The optimal $\mathbf{T}$ can be found in closed-form as $\mathbf{T} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$. If $d_x > n$ the matrix $\mathbf{X}^T\mathbf{X}$ will be rank deficient. In this situation dimensionality reduction or regularization is often necessary. A common approach is to independently learn low-dimensional

[1] Bold capital letters denote matrices $\mathbf{X}$, bold lower-case letters a column vector $\mathbf{x}$. $\mathbf{x}_j$ represents the j^{th} column of the matrix $\mathbf{X}$. All non-bold letters represent scalar variables. x_{ij} denotes the scalar in the row i and column j of the matrix $\mathbf{X}$ and the scalar i^{th} element of a column vector $\mathbf{x}_j$. $\|\mathbf{x}\|_2^2$ denotes the $L2$-norm of the vector $\mathbf{x}$. $tr(\mathbf{A}) = \sum_i a_{ii}$ is the trace of the matrix $\mathbf{A}$ and $diag(\mathbf{a})$ denotes an operator that generates a diagonal matrix with the elements of the vector $\mathbf{a}$. $\|\mathbf{A}\|_F^2 = tr(\mathbf{A}^T\mathbf{A}) = tr(\mathbf{A}\mathbf{A}^T)$ designates the Frobenious norm of matrix $\mathbf{A}$.

models for each data set using PCA/KPCA, and then learn a linear or nonlinear relation between projections using any supervised learning technique (e.g. neural networks). Applying PCA/KPCA separately to each set preserves the directions of maximum variance within sets, but these do not necessarily correspond to the direction of maximum covariation between sets [26]. That is, independently learning low-dimensional models may result in a loss of important details relevant to the coupling between sets. The RRR model [23,24,25] finds a linear mapping, $\mathbf{T} \in \Re^{d_x \times d_y}$, that minimizes the LS error subject to rank constraints on $\mathbf{T}$, effectively reducing the number of free parameters to estimate. The RRR model minimizes $||\mathbf{Y} - \mathbf{TX}||_F^2$ subject to $rank(\mathbf{T}) = k$. A mathematically convenient way to impose $rank(\mathbf{T}) = k$ is to explicitly factorize $\mathbf{T} = \mathbf{BA}^T$, where $\mathbf{A} \in \Re^{d_\varphi \times k}$ and $\mathbf{B} \in \Re^{d_y \times k}$ are regression matrices, and k denotes the rank of the reduced rank model.

The Kernel RRR (KRRR) model minimizes the following energy function:

$$\min_{\mathbf{A},\mathbf{B}} \|\mathbf{Y} - \mathbf{BA}^T\varphi(\mathbf{X})\|_F^2, \tag{1}$$

where $\varphi(\cdot)$ is a nonlinear function that transforms $\mathbf{X}$ to a (usually) high-dimensional feature space. The surface of Eq. (1) has a unique minimum, up to an invertible $k \times k$ affine transformation [27].

3.2 Learning Parameters for KRRR

This section explores three numerical schemes to optimize Eq. (1). The three methods are the Matlab function eigs to solve Generalized Eigenvalue Problems (GEPs), the Subspace Iteration (SI) method, and Alternated Least-Squares (ALS) procedure. We compare the computational cost as well as the error achieved by the algorithms.

1-Matlab Eigs function (EIGS): Without loss of generality the matrix $\mathbf{A}$ in Eq. (1) can be expressed as a linear combination of $\varphi(\mathbf{X})$, i.e. $\mathbf{A} = \varphi(\mathbf{X})\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} \in \Re^{n \times k}$. $\mathbf{K} = \varphi(\mathbf{X})^T\varphi(\mathbf{X})$ is the kernel matrix such that each entry $k_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \langle\varphi(\mathbf{x}_i), \varphi(\mathbf{x}_j)\rangle$ measures the similarity between two samples by means of a kernel function. Optimizing over $\mathbf{B}$ (i.e. $\mathbf{B} = \mathbf{YK}\boldsymbol{\alpha}(\boldsymbol{\alpha}^T\mathbf{K}^2\boldsymbol{\alpha})^{-1}$) and substituting the optimal $\mathbf{B}$ value in Eq. (1) results in the following minimization w.r.t $\boldsymbol{\alpha}$:

$$\min_{\boldsymbol{\alpha}} tr\left\{(\boldsymbol{\alpha}^T\mathbf{K}^2\boldsymbol{\alpha})^{-1}(\boldsymbol{\alpha}^T\mathbf{KY}^T\mathbf{YK}\boldsymbol{\alpha})\right\}. \tag{2}$$

Solving $\boldsymbol{\alpha}$ is a GEP, and we used the Matlab eigs function. Once $\boldsymbol{\alpha}$ is known, $\mathbf{B} \in \Re^{d_y \times k}$ can be computed with standard regression as:

$$\mathbf{B} = \mathbf{YK}\boldsymbol{\alpha}(\boldsymbol{\alpha}^T\mathbf{K}^2\boldsymbol{\alpha})^{-1}. \tag{3}$$

2-Subspace Iteration (SI): The SI method [28] is an extension of the Power method to solve GEPs. Given two symmetric matrices, $\mathbf{S}_1 \in \Re^{n \times n}$ and $\mathbf{S}_2 \in \Re^{n \times n}$, and an initial random matrix $\boldsymbol{\alpha}_0 \in \Re^{n \times k}$, the SI method [28] alternates the following steps:

$$\mathbf{S}_1\hat{\boldsymbol{\alpha}}_{t+1} = \mathbf{S}_2\boldsymbol{\alpha}_t \tag{4}$$

$$\mathbf{S} = \hat{\boldsymbol{\alpha}}_{t+1}^T\mathbf{S}_1\hat{\boldsymbol{\alpha}}_{t+1} \qquad \mathbf{T} = \hat{\boldsymbol{\alpha}}_{t+1}^T\mathbf{S}_2\hat{\boldsymbol{\alpha}}_{t+1} \tag{5}$$

$$\mathbf{SW} = \mathbf{TW}\boldsymbol{\Delta} \tag{6}$$

$$\hat{\boldsymbol{\alpha}}_{t+1} = \hat{\boldsymbol{\alpha}}_{t+1}\mathbf{W} \qquad \hat{\boldsymbol{\alpha}}_{t+1} = \hat{\boldsymbol{\alpha}}_{t+1}/||\hat{\boldsymbol{\alpha}}_{t+1}||_F.$$

where t denotes the iteration step. In our case, $\mathbf{S}_1 = \mathbf{K}^2$ and $\mathbf{S}_2 = \mathbf{K}\mathbf{Y}^T\mathbf{Y}\mathbf{K}$. The first step, Eq. (4), of the SI algorithm solves a linear system of equations to find $\hat{\boldsymbol{\alpha}}_{t+1}$. In the second step, Eq. (5), the data is projected onto the estimated subspace. In order to impose the constraints that $\boldsymbol{\alpha}_{t+1}^T\mathbf{S}_1\boldsymbol{\alpha}_{t+1} = \boldsymbol{\Lambda}$ and $\boldsymbol{\alpha}_{t+1}^T\mathbf{S}_2\boldsymbol{\alpha}_{t+1} = \mathbf{I}_k$, a normalization is done by solving the following $k \times k$ generalized eigenvalue problem, $\mathbf{S}\mathbf{W} = \mathbf{T}\mathbf{W}\boldsymbol{\Delta}$, Eq. (6), where $\mathbf{W} \in \Re^{k\times k}$ is the eigenvector matrix. It can be shown [28] that as t increases, $\boldsymbol{\alpha}_{t+1}$ will converge to the eigenvectors of problem (2) and $\boldsymbol{\Delta}$ to the eigenvalues. The convergence is achieved when $\frac{|\delta_t^{k+1} - \delta_t^k|}{\delta_t^{k+1}} < \epsilon \ \forall i$, where δ_i^k denotes the k^{th}-largest generalized eigenvalue.

3-Alternated Least Squares (ALS): The ALS algorithm alternates between fixing $\boldsymbol{\alpha}$ and solving for $\mathbf{B}$ with Eq. (3), and fixing $\mathbf{B}$ and solving for $\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} = \mathbf{K}^{-1}\mathbf{Y}^T\mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}$.

For all methods we used probabilistic PCA to factorize the matrix $\mathbf{K}$ as $\mathbf{K} \approx \mathbf{U}\mathbf{S}\mathbf{U}^T + \sigma^2\mathbf{I}_n$. This factorization is beneficial to regularize the solution and make some algorithms more efficient (e.g. solving Eq. 4). See [29] for more information.

Comparison of EIGS, SI and ALS

To evaluate the computational complexity and accuracy of the three approaches to compute the parameters in KRRR, we used 50% of the subjects from session 1 in the CMU Multi-PIE [30] database as training set. The neutral and smiling faces were used for training. We used a Gaussian kernel and the local bandwidth is selected as the mean pair-wise distance. The dimension of the images is $d_y = 35999$ pixels. The number of people $n = 125$, and k is set to $k = 37$, that preserves 99.9% of the $\mathbf{K}^2$ energy (an upper bound on the rank of the GEP).

The performance is measured using the Gradient Mean Square Error (GMSE) [12]:

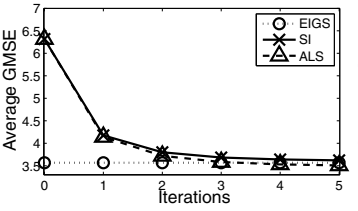
$$\text{GMSE} = \frac{1}{rc}\sum_{i=1}^{rc}\left\| \begin{bmatrix} \nabla\mathbf{F}_x(i) \\ \nabla\mathbf{F}_y(i) \end{bmatrix}_{true} - \begin{bmatrix} \nabla\mathbf{F}_x(i) \\ \nabla\mathbf{F}_y(i) \end{bmatrix}_{syn} \right\|_F^2, \tag{7}$$

between the synthesized expression and the ground truth image, where $\mathbf{F} \in \Re^{r\times c}$ is the face image of size $r \times c$ pixels, $[\mathbf{F}(i)]_{syn}$ and $[\mathbf{F}(i)]_{true}$ represent the gray level of the i^{th} pixel in the synthesized expression and the ground truth image respectively. $\begin{bmatrix} \nabla\mathbf{F}_x(i) \\ \nabla\mathbf{F}_y(i) \end{bmatrix}$ is the gradient at the i^{th} pixel. GMSE measures the difference in gradients.

Fig. 2 (a) shows the average GMSE (Eq. (7)) over 125 training subjects. As shown in Fig. 2 (a), all methods achieve similar errors. Table 2 (b) shows the computational complexity to compute $\boldsymbol{\alpha}$ using the Matlab function eigs (EIGS), the SI and ALS procedure. The time in seconds on a PC with 2.2GHz CPU was $0.077s$, 0.036s, and 1.100s for EIGS, SI and ALS respectively. The SI method achieved comparable accuracy and was more computationally efficient than ALS or eigs from Matlab.

3.3 FES with KRRR

This section shows experimental results using KRRR for FES. Given a neutral face of an untrained subject $\mathbf{x}_t$, we can synthesize a new facial expression $\mathbf{y}_t$ as a linear combination of facial expressions from the training set (i.e. $\mathbf{Y}$):



(a)

EIGS	SI	ALS
$O(n^3)$	$O(n^2k)$	$O(n^2d_y)$

(b)

Fig. 2. (a) Average GMSE to compute parameters of KRRR using Matlab eigs function (EIGS), SI, and ALS. **(b)** Computational complexity to compute $\boldsymbol{\alpha}$ using the EIGS, SI, and ALS methods.

$$\mathbf{y}_t \approx \mathbf{B}\boldsymbol{\alpha}^T\mathbf{k}(\cdot,\mathbf{x}_t) = \mathbf{Y}\mathbf{K}\boldsymbol{\alpha}(\boldsymbol{\alpha}^T\mathbf{K}^2\boldsymbol{\alpha})^{-1}\boldsymbol{\alpha}^T\mathbf{k}(\cdot,\mathbf{x}_t) = \mathbf{Y}\mathbf{g}_t, \tag{8}$$

where $\mathbf{g}_t = \mathbf{K}\boldsymbol{\alpha}(\boldsymbol{\alpha}^T\mathbf{K}^2\boldsymbol{\alpha})^{-1}\boldsymbol{\alpha}^T\mathbf{k}(\cdot,\mathbf{x}_t) \in \Re^{n\times 1}$ is the coefficient that weights the contributions of each training sample. $\mathbf{k}(\cdot,\mathbf{x}_t) \in \Re^{n\times 1}$ is the column vector of the kernel between the training samples and $\mathbf{x}_t$.

Note that in Eq. (8), the overall pixel intensity of $\mathbf{y}_t$ depends on the elements of the kernel vector $\mathbf{k}(\cdot,\mathbf{x}_t)$, which are close to 1 when $\mathbf{x}_t$ is close to the training data $\mathbf{X}$. However, the kernel values are smaller than 1 when $\mathbf{x}_t$ is far away. To normalize the kernel (i.e. $\sum_{j=1}^{n} g_{tj} \approx 1$), we use the Soft-Max kernel [31]:

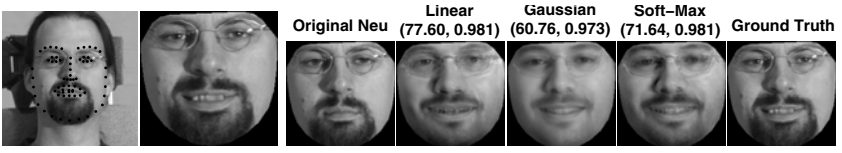
$$k(\mathbf{x}_i,\mathbf{x}_j) = \frac{exp\left(\frac{-\|\mathbf{x}_i-\mathbf{x}_j\|_2^2}{\sigma^2}\right)}{\sum_l exp\left(\frac{-\|\mathbf{x}_l-\mathbf{x}_j\|_2^2}{\sigma^2}\right)}, \quad i,j = 1,\cdots,n. \tag{9}$$

Fig. 3 shows an example of smiling facial expression synthesis using KRRR on subjects from session 1 (249 subjects) of the CMU Multi-PIE [30]. We used 50% of the subjects for training and the remaining for testing. All selected faces have been manually labeled with 66 landmarks and warped to a normalized template (see Fig. 3 (a)). The warping was done by interpolating the triangular meshes between the original landmarks and the canonical template. Note that the wrapping alone cannot result in realistic synthesis of expressions because it cannot model appearance changes (e.g. wrinkles and teeth). We compared the synthesis capabilities for three kernels: linear, Gaussian, and Soft-max. We provided two measures of error between the synthesized expression (syn) and the ground truth image (true): the average Gradient Mean Square Error (GMSE) defined in Eq. (7) and the Normalized Inner-Product (NIP):

$$\text{NIP} = \frac{1}{rc}\frac{\sum_{i=1}^{rc}[\mathbf{F}(i)]_{true}[\mathbf{F}(i)]_{syn}}{\|[\mathbf{F}]_{true}\|_F\,\|[\mathbf{F}]_{syn}\|_F}. \tag{10}$$

GMSE measures the difference in gradients, while NIP measures the correlation between gray-level values.

As can be seen in Fig. 3 the Soft-Max kernel synthesized more photo-realistic images being able to reproduce the teeth while preserving the facial hair. It also achieved the higher NIP value.



(a) (b)

Fig. 3. (a) 66 facial landmarks and the geometrically normalized face. **(b)** Synthesizing smiling faces with KRRR. Neutral test image, linear kernel, Gaussian kernel, Soft-Max kernel, and ground truth. The first number in the brackets indicates the average GMSE and the second represents the average NIP, defined in Eq. (7) and (10) respectively.

4 Bilinear Kernel Reduced Rank Regression

In the previous section, we have shown how LRRR and KRRR can be used for FES. However, observe that RRR and KRRR are unsuccessful in preserving details of the original images (e.g. wrinkles, pimples, glasses). This is because the synthesized image is a combination of the training set images, and in the training set many of these features are not present (see Fig. 3). In this section, we propose to use Bilinear KRRR (BKRRR) to effectively decouple identity and expression factors by enforcing the same identity in the synthesis of different expressions. The BKRRR is able to preserve person-specific facial features and greatly improve the synthesis performance.

4.1 Error Function for BKRRR

Let $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \in \Re^{d_x \times n}$ be a matrix containing the d_x dimensional input vectors representing neutral faces for n different subjects, and $\mathbf{Y}_l = \{\mathbf{y}_1, \cdots, \mathbf{y}_n\} \in \Re^{d_y \times n}$ be a matrix containing the vectorized images of the same n subjects with the l^{th} expression ($l = 1, \cdots, r$) (e.g. smile, surprise, disgust, squint, and scream). BKRRR extends KRRR, Eq. (1), by minimizing:

$$E(\boldsymbol{\alpha}, \mathbf{B}_l^{Exp}, \mathbf{B}^{Neu}) = \sum_{l=1}^{r} \|\mathbf{Y}_l - \mathbf{B}_l^{Exp}\boldsymbol{\alpha}^T\mathbf{K}\|_F^2 + \|\mathbf{X} - \mathbf{B}^{Neu}\boldsymbol{\alpha}^T\mathbf{K}\|_F^2, \quad (11)$$

recall that $\mathbf{A} = \varphi(\mathbf{X})\boldsymbol{\alpha}$ and it represents the space of identity, while $\mathbf{B}^{Neu}$ is a basis to reconstruct neutral faces and $\mathbf{B}_l^{Exp}$ is a basis for reconstructing the l^{th} facial expression. Unlike KRRR, BKRRR seeks to approximate all r expressions and the neutral face with the same identity coefficients. Observe that reconstructing the neutral testing image (second term in Eq. 11) will be a key component of our algorithm to decide which person-specific features will be able to be reconstructed as a combination of the training set. $\mathbf{K} \in \Re^{n \times n}$ is the kernel matrix containing the similarity between the neutral faces in the training samples. $\boldsymbol{\alpha}$, $\mathbf{B}^{Neu}$ and $\mathbf{B}_l^{Exp}$ are respectively computed as:

$$\min_{\boldsymbol{\alpha}} tr\left\{(\boldsymbol{\alpha}^T\mathbf{K}^2\boldsymbol{\alpha})^{-1}\left[\boldsymbol{\alpha}^T\mathbf{K}\left(\sum_{l=1}^{r}\mathbf{Y}_l^T\mathbf{Y}_l + \mathbf{X}^T\mathbf{X}\right)\mathbf{K}\boldsymbol{\alpha}\right]\right\},$$

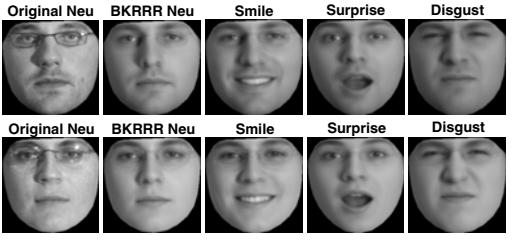


Fig. 4. Synthesis of facial expressions with BKRRR. First column shows the input image, the second the synthesized neutral image, the third, fourth and fifth show the synthesized smile, surprise and disgust expression respectively. Observe that BKRRR can not reconstruct the glasses.

$$\mathbf{B}^{Neu} = \mathbf{X}\mathbf{K}\boldsymbol{\alpha}(\boldsymbol{\alpha}^T\mathbf{K}^2\boldsymbol{\alpha})^{-1}, \tag{12}$$

$$\mathbf{B}_l^{Exp} = \mathbf{Y}_l\mathbf{K}\boldsymbol{\alpha}(\boldsymbol{\alpha}^T\mathbf{K}^2\boldsymbol{\alpha})^{-1}, \quad (l = 1, \cdots, r). \tag{13}$$

Similar to Section 2, solving $\boldsymbol{\alpha}$ is a GEP and we use the SI method.

The matrix $\boldsymbol{\Theta} = \boldsymbol{\alpha}^T\mathbf{K} \in \Re^{k\times n}$ in BKRRR contains subspace of identity variation. Given a new testing image $\mathbf{x}_t$, the synthesized expression can be obtained as:

$$\mathbf{y}_t = \mathbf{B}_l^{Exp}\boldsymbol{\alpha}^T\mathbf{k}(\cdot, \mathbf{x}_t), \tag{14}$$

where $\mathbf{k}(\cdot, \mathbf{x}_t)$ is the kernel vector for $\mathbf{x}_t$. Similarly, for the neutral face:

$$\mathbf{x}_t^{Neu} = \mathbf{B}^{Neu}\boldsymbol{\alpha}^T\mathbf{k}(\cdot, \mathbf{x}_t), \tag{15}$$

which approximates the neutral expression of the testing sample using the training data (2^{nd} column of Fig. 4). The synthesis of the neutral face image from the training images is important to recover subtle person-specific features and its use will be discussed in the next section. Fig. 4 also shows other synthesized expressions (smile, surprise and disgust) using the BKRRR model.

4.2 Preserving Person-Specific Features

Fig. 3 and Fig. 4 show a fundamental problem of regression approaches: the synthesized image is a combination of the data, and it is usually difficult to reconstruct subtle person-specific features of the testing image as holistic combinations of training samples. Moreover, it is not realistic to assume that the training data includes all possible iconic variations (e.g. types of glasses, beards, eyes half closed). In addition, the BKRRR minimizes a least-square error, which typically does not preserve subtle person-specific features such as pimples that might have small energy (see Fig. 3 and 4). This section shows how to combine the regression results with the synthesized neutral image to preserve subtle person-specific features.

Fig. 5 illustrates the process to preserve person-specific facial details. Given a neutral test face $\mathbf{x}_t$ (Fig. 5 (a)), we first synthesize the neutral image as a combination of the

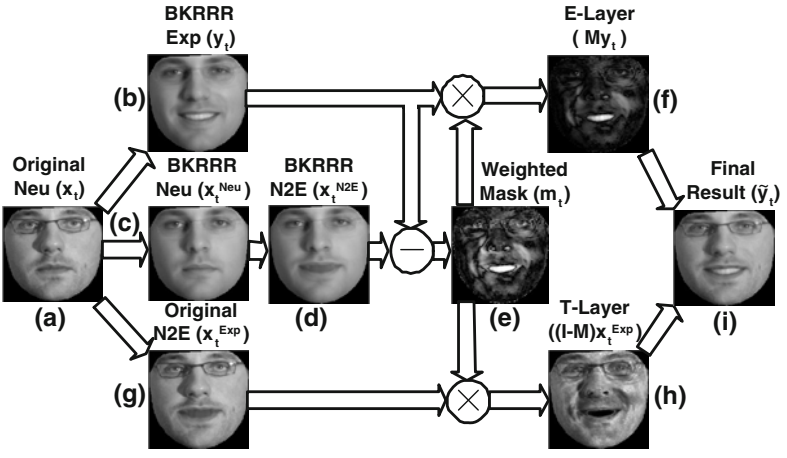


Fig. 5. FES using BKRRR that preserves person-specific facial features such as glasses and beard

training data using Eq. (15), this image is denoted as $\mathbf{x}_t^{Neu} \in \Re^{d_x \times 1}$ (Fig. 5 (c)). The resulting image $\mathbf{x}_t^{Neu}$ is warped onto the normalized template of the expression we want to target, $\mathbf{x}_t^{N2E} \in \Re^{d_y \times 1}$ (Fig. 5 (d)). We then apply BKRRR to generate $\mathbf{y}_t$ using Eq. (14) (Fig. 5 (b)). A weighted mask (Fig. 5 (e)) is computed by subtracting $\mathbf{x}_t^{N2E}$ from $\mathbf{y}_t$ as: $\mathbf{m}_t = exp\left(\frac{|\mathbf{x}_t^{N2E} - \mathbf{y}_t|}{\beta}\right)$, where $\mathbf{m}_t \in \Re^{d_y \times 1}$ denotes the weighted mask, β is a scalar selected to ensure that elements of $\mathbf{m}_t$ are between 0 and 1. The weighted mask has high values in regions where the appearance changes due to the expression variation (e.g. teeth and cheeks), and low values where the training data can not reconstruct person-specific features (e.g. glasses).

An expression layer (Fig. 5 (f)) is computed by multiplying the mask $\mathbf{M} = diag(\mathbf{m}_t) \in \Re^{d_y \times d_y}$ by the synthesized expression $\mathbf{y}_t$, that is: $\mathbf{M}\mathbf{y}_t$. This layer contains only appearance variations due to expression changes (e.g. teeth and wrinkles on the cheeks). We normalize the original neutral face $\mathbf{x}_t$ to the expression template and obtain $\mathbf{x}_t^{Exp} \in \Re^{d_y \times 1}$ (Fig. 5 (g)). Later a person-specific texture layer (Fig. 5 (h)) is created as: $(\mathbf{I} - \mathbf{M})\mathbf{x}_t^{Exp}$. Finally, the expression face $\tilde{\mathbf{y}}_t$ (Fig. 5 (i)) is computed as the combination of the expression layer and the texture layer:

$$\tilde{\mathbf{y}}_t = \mathbf{M}\mathbf{y}_t + (\mathbf{I} - \mathbf{M})\mathbf{x}_t^{Exp}. \tag{16}$$

The final result $\tilde{\mathbf{y}}_t$ greatly improves the resemblance to the original neutral test face over the result of BKRRR because it has merged person-specific features that could not be modeled by the BKRRR model.

4.3 Illumination Adaption

Fig. 6 (c) shows an example of synthesizing a smiling face using one image taken with a regular camera with uncontrolled illumination (Fig. 6 (a)). As can be observed, the

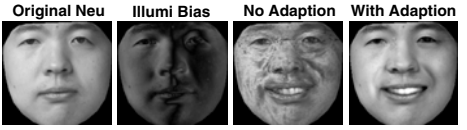


Fig. 6. Illumination normalization for FES

poor synthesis is the result of the different illumination conditions between training and testing. This section proposes a simple method to normalize illumination changes.

Fig. 6 (b) shows the illumination bias computed as the difference between the original test face and the mean face of the training set (neutral face). As can be observed, the high values of the illumination bias on the right cheek show a large difference between the training and testing lighting conditions. Fig. 6 (d) shows the results obtained after the illumination normalization.

Given the test image $\mathbf{x}_t \in \Re^{d_x \times 1}$, and the mean training face $\overline{\mathbf{x}}$, we create a representation that contains both the spatial and textural information of the image. That is, $\mathbf{F}_t = [\mathbf{l}_h, \mathbf{l}_v, \mathbf{x}_t]^T \in \Re^{3 \times d_x}$ and $\mathbf{F}^{mean} = [\mathbf{l}_h, \mathbf{l}_v, \overline{\mathbf{x}}]^T \in \Re^{3 \times d_x}$ respectively, where $[\mathbf{l}_h, \mathbf{l}_v]$ denotes the spatial location of the pixels along the horizontal and vertical axis respectively. Then we compute the linear transformation $\mathbf{M} \in \Re^{3 \times 3}$ that minimizes $||\mathbf{F}^{mean} - \mathbf{M}\mathbf{F}_t||_F^2$. The optimal matrix is $\mathbf{M} = \mathbf{F}^{mean}(\mathbf{F}_t)^+$, where $()^+$ denotes generalized pseudo-inverse. Then $\mathbf{F}_t^* = [\mathbf{l}_h, \mathbf{l}_v, \mathbf{x}_t^*]^T = \mathbf{F}^{mean}(\mathbf{F}_t)^+\mathbf{F}_t$, where $\mathbf{x}_t^*$ represents the illumination normalized testing image. Finally, to normalize the contrast of the image, the image is processed as: $\widetilde{\mathbf{x}_t} = \frac{std(\overline{\mathbf{x}})}{std(\mathbf{x})}(\mathbf{x}_t^* - mean(\mathbf{x}_t^*)) + mean(\mathbf{x}_t^*)$, where $\widetilde{\mathbf{x}_t}$ is the resulting normalized image. $std(\cdot)$ and $mean(\cdot)$ are operators that compute the standard deviation and mean respectively. Then $\widetilde{\mathbf{x}_t}$ is used to synthesize the smile expression $\widetilde{\mathbf{y}_t}$. As shown in Fig. 6 (d), the adaption algorithm greatly improves FES in images with untrained lighting conditions.

5 Experiments

This section provides quantitative and qualitative (visual) evaluation of the techniques proposed in this paper. We used all subjects (336) from the four sessions of the CMU Multi-PIE database [30]. We selected the subset of frontal faces containing 919 neutral faces, 249 smiling faces from session 1, 203 surprise faces from session 2, 203 squint faces from session 2, 228 disgust faces from session 3 and 239 scream faces from session 4 respectively. All selected faces have been manually labeled with 66 landmarks and warped to a normalized template (see Fig. 3 (a)).

5.1 FES with BKRRR

This section compares the performance of Linear Reduced Rank Regression(LRRR), KRRR, BKRRR, Tensor (HOSVD) [13,12] and BKRRRT (BKRRR with texture preservation described in section 4.2). The performance of each method is measured using GMSE (Eq. (7)) and NIP (Eq. (10)).

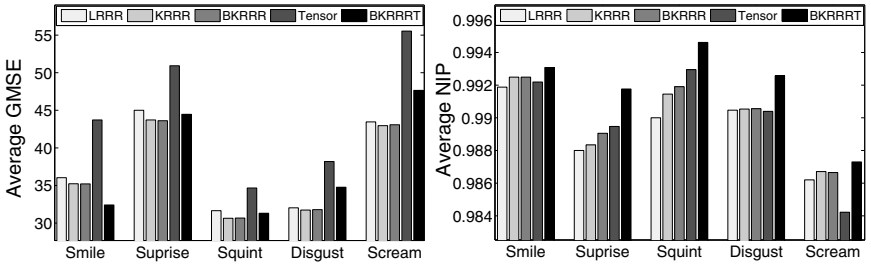


Fig. 7. Comparison of LRRR, KRRR, BKRRR, Tensor (HOSVD) and BKRRRT (BKRRR+Texture) in terms of the average GMSE (the lower the better) and average NIP (the higher the better)

Fig. 8. FES on neutral faces with subtle person-specific features (e.g. hair, wrinkle, glasses, mole and beard). The first number in brackets indicates the average GMSE and the second average NIP.

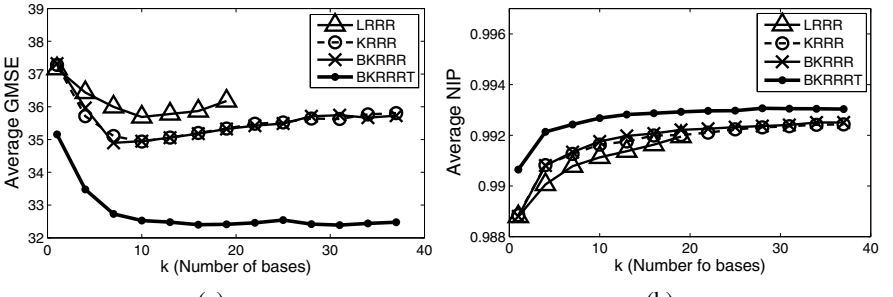
We used 50% of the faces from the CMU Multi-PIE database [30] for training (i.e. 125 for smile, 102 for surprise, 102 for squint, 114 for disgust and 120 for scream) and the remaining 50% for testing and cross-validation. In the tensor method [13], we selected all bases whose singular values are non-zero, to maximize the expressibility of

the model (as done in [13]). For LRRR, KRRR, BKRRR, and BKRRRT we selected the number of basis, k, as the number of eigenvectors that preserve 99.9% of the energy in $\mathbf{K}^2$. This is an upper bound on the rank of the RRR model. For both the KRRR and BKRRR methods, we used the Soft-Max kernel, and the regression matrices were computed using the SI method. The bandwidth parameter for the Soft-Max kernel was selected with cross-validation.

Numerical results are shown in Fig. 7. The LRRR, KRRR, BKRRR and BKRRRT methods all have smaller average GMSE than the tensor method. The BKRRR and KRRR have similar performance. However, recall that the BKRRR method is necessary to synthesize the neutral face as combination of the training samples used in the BKRRRT. The BKRRRT outperforms visually and quantitatively (in NIP) both BKRRR and KRRR. Fig. 8 shows several synthesized faces for all methods. The first column shows the original test image, the second column the neutral image, the third, fourth fifth and sixth column the synthesized image with BKRRR, LRRR, Tensor method [13] and BKRRRT respectively. Finally, the last column shows the ground truth image. Observe that BKRRRT can reconstruct much more accurately subtle facial features (e.g. glasses, skin, pimples, eyelids, hairs, mole and wrinkle) than any other method. Moreover, visually it is able to generate more photo-realistic images. On the other hand, the tensor method produces artifacts in the synthesized faces which reflects in a larger GMSE (worse preservation of edges) and smaller NIP (bad appearance matching). BKRRRT achieves the highest average NIP compared to all other methods. Observe, that occasionally the value for GMSE is higher than LRRR. This is because there is large difference in subtle edges in the original and synthesized image (e.g. rim of glasses slightly shifted), but BKRRRT achieves more photo-realistic results. Fig. 9 shows the average GMSE and NIP error versus the number of bases k to synthesize smile. As expected the error decreases w.r.t. the number of bases and BKRRRT clearly outperforms competitive approaches. For more results see [32].

5.2 FES with Illumination Adaption

This experiment tests the ability of our algorithm to handle untrained illumination conditions. Fig. 10 shows several images that have been taken with a regular camera under



(a) (b)

Fig. 9. Average GMSE **(a)** and average NIP **(b)** versus number of bases. We used 125 testing images from the session 1 in the CMU-MultiPIE.

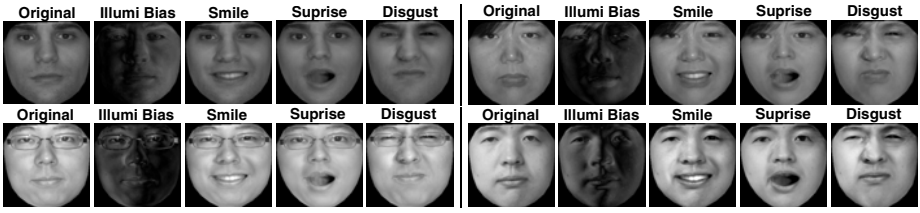


Fig. 10. FES with images taken with a regular camera under different lighting conditions. The input image is denoted by "original" and the illumination bias as "Illumi bias".

different illumination conditions. The images contain subjects of varying ethnicity. After correcting for illumination as explained in Section 4.3, our FES using BKRRRT produces very realistic results.

6 Discussion and Future Work

This paper presents a method for FES based on Bilinear KRRR. The BKRRR model learns a nonlinear mapping between a neutral face image and another image with a different facial expression of the same person. To preserve subtle person-specific features and be robust to untrained configurations, we proposed a method to combine the result of BKRRR with the original image. The results of our method are visually realistic despite the limited amount of training data. Although we have illustrated the BKRRR in the case of FES, the method is more general and can be applied to other problems image synthesis problems. In future work, we plan to improve the performance using local models (e.g. independently modeling eye, mouth and nose regions).

References

1. Gratch, J., Rickel, J., Andre, E., Cassell, J., Petajan, E., Badler, N.: Creating interactive virtual humans: some assembly required. IEEE Intelligent Systems 17, 54–63 (2002)
2. Choi, C., Aizawa, K., Harashima, H., Takebe, T.: Analysis and synthesis of facial image sequences in model-based image coding. IEEE Trans. CSVT 4, 257–275 (1994)
3. Breen, D., Lin, M.: Vision-based control of 3D facial animation. In: SCA, pp. 193–206 (2003)
4. Noh, J., Neumann, U.: Expression cloning. SIGGRAPH 1, 277–288 (2001)
5. Keeve, E., Girod, S., Kikinis, R., Girod, B.: Deformable modeling of facial tissue for craniofacial surgery simulation. Computer Aided Surgery 3, 223–228 (1998)
6. Liu, Z., Shan, Y., Zhang, Z.: Expressive expression mapping with ratio images. In Proc. of Ann. Conf. on Computer Graphics and Interactive Techniques (2001)
7. Zhang, Q., Liu, Z., Guo, B., Shum, H.: Geometry-driven photorealistic facial expression synthesis. IEEE Trans. VCG 12, 48–60 (2006)
8. Chung, K.: Gross Anatomy (Board Review). Lippincott Williams & Wilkins, Hagerstown (2005)
9. Nguyen, M., Lalonde, J., Efros, A., De la Torre, F.: Image-based shaving. Computer Graphics Forum (Eurographics) 27, 627–635 (2008)

10. Vasilescu, M., Terzopoulos, D.: Multilinear analysis of image ensembles: Tensorfaces. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 447–460. Springer, Heidelberg (2002)
11. Tenenbaum, J., Freeman, W.: Separating style and content with bilinear models. Neural Computation 12, 1247–1283 (2000)
12. Wang, H., Ahuja, N.: Facial expression decomposition. In: ICCV (2003)
13. Abboud, B., Davoine, F.: Appearance factorization for facial expression analysis. In: BMVC (2004)
14. Vlasic, D., Brand, M., Pfister, H., Popovic, J.: Face transfer with multiliner models. ACM Trans. Graphics 24, 426–433 (2005)
15. Macedo, I., Brazil, E., Velho, L.: Expression transfer between photographs through multilinear aam's. In: SIBGRAPI, pp. 239–246 (2006)
16. Bettinger, F., Cootes, T., Taylor, C.: Modelling facial behaviours. In: BMVC, vol. 2 (2002)
17. Zalewski, L., Gong, S.: Synthesis and recognition of facial expressions in virtual 3D views. In: AFGR (2004)
18. Chang, Y., Hu, C., Ferisa, R., Turk, M.: Manifold based analysis of facial expression. Image and Vision Computing 24, 605–614 (2005)
19. Kouadio, C., Poulin, P., Lachapelle, P.: Real-time facial animation based upon a bank of 3D facial expressions. In: Proc. of Computer Animation (1998)
20. Pighin, F., Szeliski, R., Salesin, D.: Resynthesizing facial animation through 3D model-based tracking. In: ICCV (1999)
21. Pyun, H., Kim, Y., Chae, W., Kang, H., Shin, S.: An example-based approach for facial expression cloning. In: SIGGRAPH/Eurographics SCA, pp. 167–176 (2003)
22. Parke, F.I., Waters, K.: Computer facial animation. AK Peters, Wellesley (1996)
23. Anderson, T.: Estimating linear restrictions on regression coefficients for multivariate normal distributions. Annals of Mathematics Statistics 12, 327–351 (1951)
24. Scharf, L.: The SVD and reduced rank signal processing. Signal Processing 25, 113–133 (2002)
25. Diamantaras, K.: Principal Component Neural Networks (Therory and Applications). John Wiley & Sons, Chichester (1996)
26. De la Torre, F., Black, M.: Dynamic coupled component analysis. In: CVPR (2001)
27. Baldi, P., Hornik, K.: Neural networks and principal component analysis: Learning from examples without local minima. Neural Networks 2, 53–58 (1989)
28. Bathe, K., Wilson, E.: Numerical Methods in Finite Element. Prentice-Hall, Englewood Cliffs (1971)
29. De la Torre, F., Gross, R., Baker, S., Kumar, V.: Representational oriented component analysis for face recognition with one sample image per training class. In: CVPR (2005)
30. Gross, R., Matthews, I., Cohn, J., Kanade, T., Baker, S.: The CMU multi-pose, illumination, and expression (multi-pie) face database. Tech. rep., Robotics Institute, Carnegie Mellon University,TR-07-08 (2007)
31. Weinberger, K., Tesauro, G.: Metric learning for kernel regression. In: AISTATS (2007)
32. Huang, D., De la Torre, F.: Bilinear kernel reduced rank regression for facial expression synthesis. Tech. rep., Robotics Institute, Carnegie Mellon University,TR-10-23 (2010)

Multi-class Classification on Riemannian Manifolds for Video Surveillance

Diego Tosato[1], Michela Farenzena[1], Marco Cristani[1,2], Mauro Spera[1], and Vittorio Murino[1,2]

[1] Dipartimento di Informatica, University of Verona, Italy
[2] Istituto Italiano di Tecnologia (IIT), Genova, Italy

Abstract. In video surveillance, classification of visual data can be very hard, due to the scarce resolution and the noise characterizing the sensors' data. In this paper, we propose a novel feature, the ARray of COvariances (ARCO), and a multi-class classification framework operating on Riemannian manifolds. ARCO is composed by a structure of covariance matrices of image features, able to extract information from data at prohibitive low resolutions. The proposed classification framework consists in instantiating a new multi-class boosting method, working on the manifold Sym_d^+ of symmetric positive definite $d \times d$ (covariance) matrices. As practical applications, we consider different surveillance tasks, such as head pose classification and pedestrian detection, providing novel state-of-the-art performances on standard datasets.

1 Introduction

An important goal of automated video surveillance is to design algorithms that can characterize different objects of interest (OIs), especially when immersed in a cluttered background and captured at low resolution. The detection (e.g., of faces or pedestrians) and the classification (e.g., of facial poses) are among the most studied applications. In the multi-faceted plethora of approaches in the literature (see [1,2,3] for extensive reviews), boosting-based techniques play a primary role [4,5,6,7,8,9,10,11,12,13,14]: boosting [15,16,17] is a remarkable, highly customizable way to create strong and fast classifiers, employing various features fed into diverse architectures, with specific policies.

Among the different features considered for boosting in object classification (see [18] for an updated list), covariance features [19] have been exploited as powerful descriptors of pedestrians [11,12,13], and their effectiveness has been explicitly investigated in a comparative study [14]. When injected in boosting systems [11,12,13,14], covariances provide strong detection performances. They encapsulate the high intra-class variances (due to pose and view changes of the OI), they are in general stable in presence of noise, and provide an elegant way to fuse multiple low-level features, as they intrinsically exploit possible inter-features' dependencies. Moreover, thanks to the integral image representation, they can by calculated in a very efficient way.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 378–391, 2010.

Since covariance matrices lie in the Riemannian manifold of symmetric positive definite matrices Sym_d^+, their usage in a boosting framework requires a careful treatment. In [11], the input covariance features are projected into the tangent space at particular points of the manifold, where an Euclidean metric can be instantiated, and the Logitboost framework can be applied.

In this paper, we propose two main contributions. First, we present a novel kind of feature, *i.e.*, the *ARray of COvariances* (ARCO), able to describe visual objects at prohibitive low resolutions (up to 5×5 pixels): it marries the dense descriptors philosophy, adopted for example in [20], with the expressivity of the covariance information. Second, we show how such features can be embedded in a multi-classification framework by boosting, extending [11] to the multi-class case. We experimentally show that Sym_d^+ has non positive curvature and in the areas where the curvature is almost flat the Euclidean metric on the tangent space at any point on the manifold is a good approximation of the Riemannian metric. Therefore, unlike [11], we map all the data in a unique tangent space, and we perform all the computations on this (Euclidean) space where a typical multi-class LogitBoost algorithm can be applied.

The experimental trials show how we outperform the current methods in two important applications for surveillance like head pose classification and pedestrian detection, without adopting complex boosting schemes such as Floatboosting for pyramids [9], decision trees [6], VectorBoosting for width-first-search trees [7], or Probabilistic Boosting Networks [21]. We fix novel state-of-the-art performances on standard databases. This encourages the embedding of our Riemannian framework in the above quoted boosting schemes. We stress also the capability of dealing with compelling image resolutions, promoting the use of ARCOs for heterogeneous applications, especially in the surveillance field.

The rest of the paper is organized as follows. Sec. 2 describes the proposed ARCO feature, and Sec. 3 depicts the proposed multi-class framework. Sec. 4 shows the experimental results on several surveillance applications, and finally we draw our conclusions in Sec. 5.

2 ARCO: ARray of COvariance Matrices

The proposed classification framework has been specifically designed to deal with low resolution images, typical of a video surveillance scenario. In such conditions, the number of features that can be extracted are relatively small, and quite unreliable. This is very challenging in problems like, for instance, head pose classification, in which the details are crucial to distinguish the different object classes. Moreover, the classifier must cope with objects (pedestrians, heads) views in a variety of light conditions. Our solution is based on two main concepts: 1) the organization of the image into a grid of uniformly spaced and overlapping patches (Fig. 1); 2) the use of covariance matrices of image features as patch descriptors, which are classified by multi-class LogitBoost on Riemannian manifolds. In a few words, each patch classifier votes for a class, and the final classification result is the class voted by the majority of them.

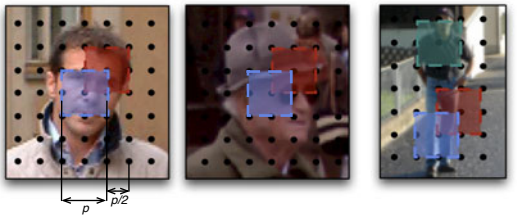


Fig. 1. Array of Covariance matrices (ARCO) feature. The image is organized as a grid of uniformly spaced and overlapping patches. On each patch, a multi-class classifier is estimated.

In [11], where the use of covariance matrix descriptors is tailored for pedestrian detection, LogitBoost was used for both a greedy estimation of the most discriminative patches among a set of different sizes and positions, and for classifying them, i.e., as feature selection and classification method at the same time. The same reasoning, using boosting for feature selection and classification, has been applied to other approaches in the literature, as for example in [22,23]. Here, instead, a feature selection operation is unfeasible, because low resolution images contain such scarce and noisy information that the result would be unreliable: it is more convenient to use *all* features in a suitable way. Our approach takes inspiration from the literature on dense image descriptors (see [20] as an example). We sample the image I into uniformly distributed and overlapping patches of the same dimension. Each patch is described by the covariance matrix representation, that encodes the local shape and appearance of the (small) region. We use these patches in a democratic way: we exalt their discriminative power by boosting a strong multi-class classifier, and we collect their classification results.

More formally, given a set of patches $\{P_i\}_{i=1,\ldots,N_P}$, we learn a multi-class classifier for each patch $\{F_{P_i}\}_{i=1,\ldots,N_P}$ through the multi-class LogitBoost algorithm [17], adapted to work on Riemannian manifolds.

Let $\Delta_j = \sum_{i=1}^{N_P}(F_{P_i} == j)$ be the number of patches that vote for the class $j \in \{1,\ldots,J\}$. We assign a class label c to an image, estimating

$$c = \arg\max_j \{\Delta_j\}, \quad j = 1,\ldots,J. \tag{1}$$

In order to increase robustness to local illumination variations, we apply the normalization operator introduced in [11] before applying the multi-class framework.

The ARCO representation has several advantages. First, it allows to take into account different features, inheriting their expressivity, and exploiting possible correlations. In this sense, it is as a compact and powerful integration of features. Second, due to the use of integral images, ARCO is fast to compute, making it suitable for a possible real-time usage. Finally, as a dense representation, it is robust to occlusions. We will prove all the above characteristics during the experimental trials in Sec.4.

3 Multi-class Classification on Riemannian Manifolds

Let $C_1, C_2, \ldots, C_J$ be the data classes whose elements (the covariances) live in the Riemannian manifold $\mathcal{M}$ of $d \times d$ symmetric positive definite matrices denoted by Sym_d^+. Let $\mathcal{S} = \{X_i, y_i\}_{i=1,\ldots,N}$ be the set of N training examples, with $X_i \in \mathcal{M}$ and label $y_i \in \{1, \ldots, J\}$. The goal is to produce a function $F(X_i) : \mathcal{M} \mapsto \{1, \ldots, J\}$ as

$$F(X_i) = \arg\max_j \{F_j(X_i)\}, \quad j = 1, \ldots, J. \tag{2}$$

F_j is a *single-class* strong classifier, and it is defined, in turn, as a sum of L weak classifiers $\{f_{lj}\}_{l=1,\ldots,L}$. These weak classifiers are learned by multi-class LogitBoost.

3.1 Riemannian Geometry on Sym_d^+

In this section, we briefly review the geometry of Sym_d^+, the manifold consisting of all $d \times d$ symmetric definite positive matrices (covariance matrices), extending the treatment given in [11].

The tangent space T_Y at any point $Y \in Sym_d^+$ can be identified with Sym_d, the (vector) space of $d \times d$ symmetric matrices.

The mapping of X on T_Y, is given by the point-dependent $\log_Y$ operator:

$$\log_Y(X) = Y^{\frac{1}{2}} \log\left(Y^{-\frac{1}{2}} X Y^{-\frac{1}{2}}\right) Y^{\frac{1}{2}}, \tag{3}$$

inverse to the exponential map.

The (geodesic) distance on Sym_d^+ is defined as

$$d^2(X_1, X_2) = Tr(\log(X_1^{-\frac{1}{2}} X_2 X_1^{-\frac{1}{2}})^2) = \sum_{i=1}^{d} (\log \xi_i)^2 \tag{4}$$

where the ξ_i's are the (positive) eigenvalues of $X_1^{-\frac{1}{2}} X_2 X_1^{-\frac{1}{2}}$.

On the tangent space, the Euclidean distance

$$d_{\mathcal{E}}^2(x_1, x_2) = Tr[(x_1 - x_2)^2], \tag{5}$$

with $x_1 = \log_Y X_1$ and $x_2 = \log_Y X_2$ for any $Y \in Sym_d^+$, is the first-order approximation of Eq. (4).

In [11], a boosting framework on Sym_d^+ for detection (*i.e.*, binary classification) is presented. The idea is to build weak learners by regression over the mappings of the training points on a suitable tangent plane. This tangent plane is defined over the weighted Karcher mean [24] of the positive training data points, such to preserve their local layout on Sym_d^+. The negative points, instead, (i.e., all but pedestrians) are assumed to be spread on the manifold, thus including them in the mean estimation would bias the result. Once moving from binary to multi-class classification the above considerations do not hold anymore, because we have many "positive" classes, each of them localized in a different part of

the manifold. Therefore, 1) choosing the Karcher mean of one class would privilege that class with respect to the others, 2) the Karcher mean of all classes is inadequate.

A thorough analysis of Sym_d^+ opens a new perspective. First, its *sectional curvature*, the natural generalization of the classical Gaussian curvature for surfaces, is non-positive. Since Sym_d^+ is actually a symmetric space, the following formula holds for computing the sectional curvature κ_{I_d} at I_d – due to the homogeneity of Sym_d^+ [25], there is no loss of generality – with x,$y \in Sym_d$ linearly independent:

$$\kappa_{I_d}(x,y) = \frac{\langle R(x,y)x,y\rangle}{\| x \|^2 \| y \|^2 - \langle x,y\rangle^2} = \frac{Tr([[x,y],x]y)}{Tr(x^2)Tr(y^2) - (Tr(xy))^2} = \\ = 2\,\frac{Tr((xy)^2 - x^2y^2)}{Tr(x^2)Tr(y^2) - (Tr(xy))^2}, \tag{6}$$

by the cyclical property of the trace. Here, $[x,y] = xy - yx$ is the matrix commutator, and $R(x,y) : z \mapsto [[x,y],z]$ is the *Riemann curvature operator* (in the symmetric space framework). It can be shown (for the actual proof, see Appendix in the additional material), that $\kappa_{I_d}(x,y) \leq 0$.

Now, an application of Preissmann's theorem [25] shows that, taking the geodesic triangle with vertices I_d, X_1, X_2, one gets

$$d_{\mathcal{E}}(\log_{I_d} X_1, \log_{I_d} X_2) \leq d(X_1, X_2) \tag{7}$$

More precisely,

$$d(X_1, X_2) = d_{\mathcal{E}}(\log_{I_d} X_1, \log_{I_d} X_2) + \Xi(\kappa_{I_d}) \tag{8}$$

where $\Xi(\kappa_{I_d}) \geq 0$ is a function that depends on the sectional curvature. An explicit form for Ξ cannot be easily derived, but it is evident that if the sectional curvature is "small", one can replace the "true" distance by the Euclidean one.

Notice that the above remark reconcile the present "classical" approach with the one in [26,27], where the Log-euclidean metric is employed throughout, upon endowing Sym^+ with a Lie group structure.

The reasoning above suggests a practical manoeuvre to check this condition. We randomly pick a representative set of covariance matrices from the datasets under observation and we estimate the sectional curvature (Eq. 6) for each pair, calculating the mean at the end. Experimentally, this mean value results -10^{-3}, that is far from the standard negative curvature of -1.

In this conditions, one can choose any point on Sym_d^+ on which to map the dataset, and execute the learning on that (Euclidean) space. In practice, we choose the identity matrix I_d, as this simplifies the computation. Indeed, Eq. (3) becomes

$$\log_{I_d}(X) = \log(X) = U \log(D) U^T, \tag{9}$$

where $U \log(D) U^T$ is the eigenvalue decomposition of X, with X a generic point in Sym_d^+, U is an orthogonal matrix, and $\log(D)$ is the diagonal matrix composed by the eigenvalues' logarithms.

Moreover, the tangent space is the space of symmetric matrices, but there are only $d(d+1)/2$ independent coefficients, which are the upper triangular or lower triangular part of the matrix. Thus, by applying the vector operator, an orthonormal coordinate system for the tangent space is defined as

$$\mathrm{vec}_{I_d}(x) = \mathrm{vec}(x) = [x_{1,1}\ x_{1,2}\ \ldots x_{1,d}\ x_{2,2}\ x_{2,3}\ \ldots x_{d,d}], \tag{10}$$

where x is the map of $X \in Sym_d^+$ in the tangent space. This operator relates the Riemannian metric on the tangent space to the canonical metric defined in $\mathbb{R}^m$, with $m = d(d+1)/2$.

3.2 Algorithm Description

Following the considerations above, we map our dataset $\mathcal{S}$ to the tangent Euclidean space T_{I_d}, and we perform the classification directly on this space. In this way, $\mathcal{S}_T = \{\mathbf{x}_i, y_i\}_{i=1,\ldots,N}$ is the mapped dataset, with $\mathbf{x}_i = \mathrm{vec}(\log_{I_d}(X_i))$.

The essence of a boosting algorithm is an iterative re-weighting system that tends to focus on the most difficult examples in the training set. In the multi-class classification there are J different sets of weights built from the posterior distribution. Let $\Pr_j[\mathbf{x}_i]$ be the posterior probability for a training example $\mathbf{x}_i$ to belong to the j-th class. It is computed as:

$$\Pr_j[\mathbf{x}_i] = \frac{e^{F_j(\mathbf{x}_i)}}{\sum_{k=1}^{J} e^{F_k(\mathbf{x}_i)}}, \quad F_j(\mathbf{x}_i) = \sum_{l=1}^{L} f_{lj}(\mathbf{x}_i), \tag{11}$$

where $\{f_{lj}\}_{l=1,\ldots,L}$ is a class-specific set of weak learners. Each example in the training set $\mathcal{S}_T$ is associated to a weight that depends on the class considered:

$$w_{ij} = \Pr_j[\mathbf{x}_i](1 - \Pr_j[\mathbf{x}_i]). \tag{12}$$

The core of the learning process is the definition of the inter-class decision boundaries, which is carried out by weak learners. We build weak classifiers $g_{lj} : T_{I_d} \mapsto \mathbb{R}$ that solve a binary problem, one class against the others, then the multi-class classifiers $f_{lj} : T_{I_d} \mapsto \mathbb{R}$ derive from their combination.

The binary weak learners g_{lj} solve a weighted regression problem, whose goodness of fit is measured by the response values z_{ij}, defined as:

$$z_{ij} = \frac{y_{ij}^* - \Pr_j[\mathbf{x}_i]}{\Pr_j[\mathbf{x}_i](1 - \Pr_j[\mathbf{x}_i])}, \tag{13}$$

where $y_{ij}^* = (j == y_i)$. The combination of a set of J binary weak learners g_{lj} is provided by the following equation [17]:

$$f_{lj}(\mathbf{x}_i) = \frac{J-1}{J}\left(g_{lj}(\mathbf{x}_i) - \frac{1}{J}\sum_{k=1}^{J} g_{lk}(\mathbf{x}_i)\right). \tag{14}$$

Please note that this operation is possible because the $g_{lk}(\cdot)$s live in the same domain T_{I_d}. If the binary classification had been carried out mapping each class

in a different space, similarly to [11], the combination of the results would have been much more complicated and unclear. Working on $T_{\mathbf{Id}}$ represents an elegant and reasonable solution to the problem.

In the following we explain some details of the algorithm, summed up in pseudo-code here below.

Algorithm 1. Multi-class LogitBoost on $\mathcal{M}$

Require: $(X_1, y_1), \ldots, (X_N, y_N)$ with $X_i \in \mathcal{M}$ e $y_i \in \{1, \ldots, J\}$
- Map the data points to the tangent space T_{I_d}, by $\mathbf{x}_i = (\log_{I_d}(X_i))$
- Start with weights $w_{ij} = 1/N$ and $i = 1, \ldots, N$, $F_j(\mathbf{x}_i) = 0$ e $\Pr_j[\mathbf{x}_i] = 1/J \ \forall j$.
for $l = 1, 2, \ldots, L$ **do**
  for $j = 1, 2, \ldots, J$ **do**
    - Compute the response values (Eq. 12) and weights (Eq. 13).
    - Fit the function $g_{lj}(\mathbf{x}_i) : \mathbb{R}^m \mapsto \mathbb{R}$ by weighted least-square regression of z_{ij} to $\mathbf{x}_i$ using weights w_{ij}.
    - Set $F_j(\mathbf{x}_i) \leftarrow F_j(\mathbf{x}_i) + f_{lj}(\mathbf{x}_i)$ where $f_{lj}(\mathbf{x}_i)$ is defined in Eq. (14).
    - Update $\Pr_j[\mathbf{x}_i]$ as in Eq. (11).
    - Save $F_j = \{g_{lj}\}$.
  end for
end for
- Save the ensemble of classifiers $\{F_1, \ldots, F_J\}$.

3.3 Algorithm Details

Binary weak classification strategy. In boosting, it is possible to use very different types of weak learners. The most common are the decision stumps (or regression stumps), which are piecewise constant regression functions or linear regression functions. The original LogitBoost algorithm adopts linear regression functions as proposed in [17]. In a binary classification task a linear regression can be sufficient to solve the problem, as shown in [11] for pedestrian detection. However, a more powerful weak classification strategy is mandatory for the multi-class classification problem, as evidenced in [21], where piecewise constant functions are used.

After investigating different solutions, we have selected the weighted *regression trees* [28], which are more powerful than global models, like linear or polynomial regressors, where a single predictive formula is supposed to hold over the entire data space, and they have lower computational costs, in both the learning and testing phases. In order to avoid the risk of overtraining of the regression tree, we establish as stopping rule a minimal number τ of observations per tree leaf, experimentally estimated (see Sec. 4).

Stop condition. It is important to specify a automatic stop criterion for the learning phase. The proposed rule is a composition of two terms. The first term takes into account the accuracy with which the classes are correctly classified:

we set the maximum accuracy τ_{acc} for all the classes. The second term concerns the *learning rate*, which is the difference in accuracy between two consecutive iterations of LogitBoost. If the learning rate is less than τ_{lr} for all the classes, we assume that the boosting process has converged to its optimal solution. More formally, the learning process is stopped at the l-th iteration, when:

$$\text{acc}_l(j) \geq \tau_{\text{acc}} \ \vee \ (\text{acc}_l(j) - \text{acc}_{l-1}(j)) \leq \tau_{\text{lr}}, \ \forall j \in \{1, \ldots, J\}, \tag{15}$$

where $\text{acc}_l(j)$ counts the examples of the j-th class correctly classified at the l-th iteration. In all the experiments, τ_{acc} is set to 99% and τ_{lr} to 1%.

Multi-class detection. Our multi-class algorithm can be naturally extended to detection purposes by simply adding a class that contains background examples. It is a very large class, because it is potentially composed by all the possible images that do not contain foreground examples. For this reason, we combine the LogitBoost classifier with a *rejection cascade structure* [4].

Algorithm 1 becomes the learning procedure of each cascade level. The stop condition for a cascade level is given by Eq. (15), except for the background class that is optimized to correctly classify at least the 35% of the examples in this class, as in [11]. In practice, we order the examples in the background (BG) class, according to $\Pr_{\text{BG}}[\mathbf{x}]$. Let $\mathbf{x}_{\text{BG}}$ be the element with $(0.35N_{\text{BG}})$-th smallest probability among all the background examples. We set $th_k = F_{\text{BG}}(\mathbf{x}_{\text{BG}})$, where k is the current cascade level.

At the cascade level $(k+1)$, the BG class is first pruned using the cascade of k classifiers, rejecting the samples correctly classified as background. To obtain the desired rejection rate, the classification response for BG is redefined as $F_{\text{BG}}(\mathbf{x}) = (F_{\text{BG}}(\mathbf{x}) - th_k)$.

Computational considerations. The proposed framework inherits some of the computational characteristics of [11], where the main cost is due to SVD factorization needed for the projection of the covariance matrices on the tangent space (see Eq. 9). In our case, the presence of a unique projection point decreases the number of required SVD factorizations. This means a dramatic reduction of the computational cost in both the learning and testing phase.

4 Experiments

In this section, we show different video surveillance applications where our framework applies: head pose classification, pedestrian detection, and head detection + pose classification. In the first two cases, where comparative tests on shared databases are feasible, we outperform the relative best performances in the literature. In the third case, only qualitative results can be appreciated.

4.1 Head Pose Classification

We build a multi-class classifier for head pose classification on the 4 Pose Head Database[1]. This dataset contains head images of dimension 50×50 (see some

[1] http://www.eecs.qmul.ac.uk/~orozco/index_files/Page558.htm

samples in Fig. 2a)) obtained from the i-LIDS dataset[2]. These images come from a real video surveillance scene, mirroring well typical critical conditions: they are noisy, motion-blurred, and at low resolution. The images are divided in 4 foreground (FG) classes: Back (4200 examples), Front (3555 examples), Left (3042 examples), and Right (4554 examples). Moreover, this dataset contains another set of 2216 background (BG) images. We partition the FG dataset in 2 equal parts, using one partition for training and one for testing. We extract from each image I a set Φ of $d = 12$ features, composed by:

$$\Phi = \left[X \; Y \; R \; G \; B \; I_x \; I_y \; O \; \mathrm{Gab}_{\{0,\pi/3,\pi/6,4\pi/3\}} \right]. \tag{16}$$

X, Y represent the spatial layout in I, and R, G, B are the color channels. I_x and I_y are the directional derivatives of I, and O is the gradient orientation. Finally, Gab is a set of 4 maps containing the results of Gabor filtering. We would like to stress that these features are particularly suited for head orientation classification. Apart from the general position (X, Y) and shape information (I_x, I_y), the covariance of the color channels permits to implicitly detect hair and skin textural properties. This particularly helps in distinguishing frontal from back views. Moreover, Gabor filters emphasize facial details, such as the vertical orientation for the nose, or the horizontal orientation of the mouth, if visible. We tried different combination of these filters, and the best results are obtained with dimension 2×4, sinusoidal frequency 16, and directions $\mathcal{D} = \{0, \pi/3, \pi/6, 4\pi/3\}$. In order to give an idea on how the choice of the features affects the system's performances, Fig. 2b depicts the behavior of the system in terms of mean classification accuracy by considering different subsets of Φ. Once the features are extracted, we calculate the covariance matrices from all the patches of $p \times p$ pixels, on a fixed grid of $p/2$ pixels steps. This means that the patches remain overlapped by half of their size. We vary p, in order to investigate how the dimension (and thus, the number) of the patches modifies the classification performances. The best performance is obtained with $p = 0.32s$, where s is the image dimension. As visible in Fig. 2c, enlarging the patch dimension to more than this value diminishes the accuracy. This highlights that having a high number of small patches is better than having few large ones. This because with less, large-sized covariances all the image details are mixed together, losing the spatial information.

For each patch, a 4-class classifier is built, as described in Sec. 3.2. The τ parameter, that rules the complexity of the regression trees, has been fixed to the optimal value 150 according to the accuracy test in Fig. 2d. It is interesting to note that exceeding this value, the performance drops, which is a sign of overtraining of the system.

A very important result is the ability to maintain a high classification accuracy on extremely low resolution images. Figure 2e shows the performance of our classifier varying the image dimension s (and changing proportionally the patch parameters, with $p = \lceil 0.32s \rceil$). On a 5×5 image we reach an average accuracy above 82%.

[2] http://scienceandresearch.homeoffice.gov.uk/hosdb/cctv-imaging-technology/i-lids/

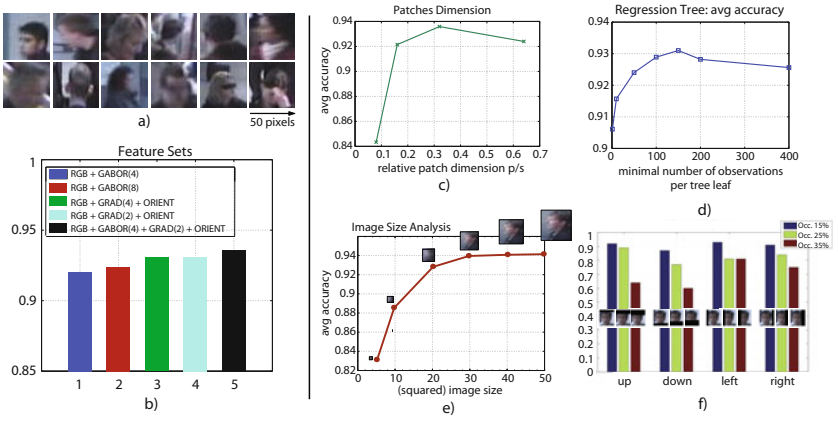


Fig. 2. Head pose classification analysis. (a) Some images from the considered dataset. Classification performance in terms of mean classification accuracy varying (b) the feature vector Φ, (c) the patch dimensions p, (d) the regression tree stop criterion (the number τ of elements per leaf), (e) the test image dimensions, and (f) considering occlusions of different strength.

Moreover, we test the ability of our classifier to deal with occlusions. Indeed, patch-based classifiers, as part-based classifiers, are naturally able to manage the presence of occlusions. We depict in Figure 2f the robustness to four types of occlusions (left-, right-, top- and bottom-side), in different sizes. As visible, top and bottom occlusions reduce the performances more, because they completely hide meaningful parts of the face.

Last, we compare our method with Orozco et at. [29], the state-of-the-art method for head pose classification for low resolution data. It is a head pose descriptor based on similarity distance maps to mean appearance templates of head images at different poses. All images in this dataset have their related pose descriptors, provided by the authors themselves [29]. The classifier is trained by Support Vector Machines (SVMs) using a polynomial kernel, as done in [29]. The result of the comparison, in terms of confusion matrix, is reported in Fig. 3. The average rate is 93.5% for our model, against 82.3% for Orozco's model.

4.2 Pedestrian Detection

We instantiate our framework on the binary problem of pedestrian detection to verify the performance of our approach on a pure detection task. We consider the INRIA Person dataset [20] for testing. It contains 1212 human images for the training part of dimension 128×64 and 1133 images for the testing part. We pick a region of interest of 50×50 at the center of the pedestrian images, that corresponds to the actual region where the pedestrian is enclosed (all positive examples come with a quite large border). Then, we use the same patch configuration described above (Sec. 4.1), but with a set of features Φ more

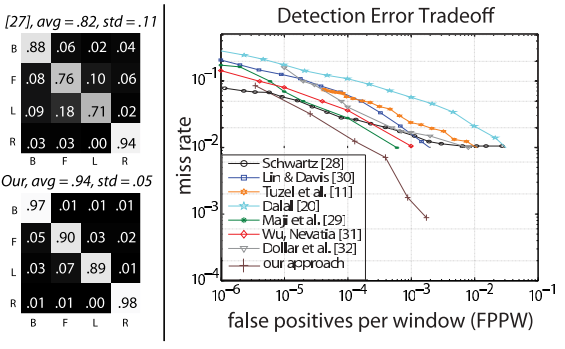


Fig. 3. Confusion matrix for the method proposed in [29] (upper left) and our method (bottom left) for head pose orientation. On the right, DET curve for pedestrian detection, compared with the state-of-the-art methods [11,20,30,31,32,33,34].

suitable for the detection task, i.e. the same proposed in [11]. In Fig. 3c, we compare our framework with [11] and with the methods in [20,30,31,32,33,34]. The performances are evaluated by the Detection Error Tradeoff (DET) curve, that expresses the proportion of true detections against the proportion of false positives, on a log-log scale. The curve is estimated by varying the threshold th_k in the range $[-1, 1]$. We use a rejection cascade of 5 levels in which each level is populated by 10000 background examples. Please, note that augmenting the number of cascade levels to more than 5 does not appreciably increase the accuracy, since the number of covariance features remains fixed (in [11], instead, at each step a new feature is selected). Our detector clearly outperforms the other methods at the state-of-the-art, especially in terms of miss-rate.

4.3 Head Pose Detection and Classification

As we are proposing a multi-class framework, we can simply add a background class to the problem at hand, to perform detection along with classification. Here, we show how the system works for the problem of head pose detection and classification.

As first experiment, we consider the 4 head pose classes of the Pose Head dataset used in Sec. 4.1, adding its 2215 background examples. We use the same optimal settings estimated above, and we compare the performance of our approach with [29]. Even though the original paper performs classification only, so the comparison is a bit unfair, their template descriptor is provided for background images as well. We add the background class to the other positive classes, and we compute the classification stage by using SVMs, as described in the paper. The comparison, shown in Fig. 4, shows the ability of our system to naturally deal with this task as well.

On the other hand, the images of this dataset, though challenging for location and scale variations, are all taken from the same scene, with scarce lighting

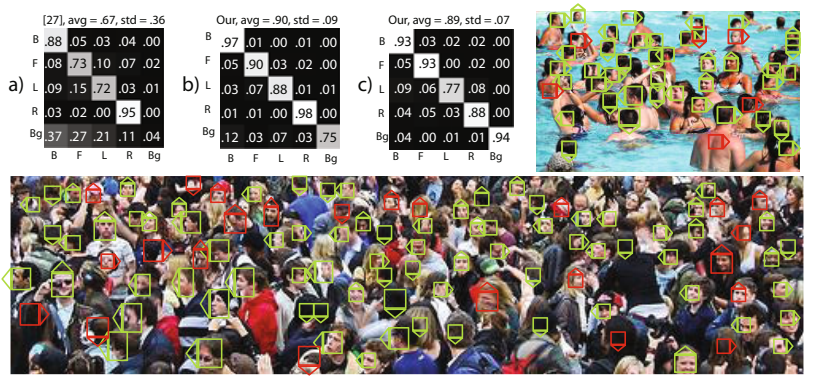


Fig. 4. Confusion matrices for the experiments on head pose detection and classification. In (a) e (b), results for the first experiment on 4 Pose Head dataset (Orozco's method in (a) [29], our method in (b)). (c) is the result of the second experiment with the more general dataset (see text for details). The other images are examples of detection and classifications in crowded scenes. The arrows indicate the head orientation. In green the correct answers provided by classifier, in red the misclassifications.

variations. Thus, the model we built is not general enough to work with different scenarios. For this reason, we perform a second experiment, building another model, and enriching the training set with new data coming from a different, more general, dataset. We use the head dataset employed in [35], composed by 2736 20×20 head images, contained in a ROI of 32×32 pixels. This dataset is mostly obtained from the INRIA person dataset, thus the images are taken from many different scenes and with a large variation of illumination conditions. The set of negative examples is composed by different real scenarios and other images containing parts of the body. We organize the data in four classes (plus background) according to heads' orientation, since the original dataset does not contain such information.

The positive examples from the 4 Pose Head dataset are resized to 20×20 pixels, whereas for the other dataset the examples are cropped from the center of the ROI. Half the data are used for training, and the testing set is composed by just the testing set of [35]. Fig. 4c summarizes the detection and classification results. Note that due to the variations in scale and position of the head, the cropped images can contain the head only partially. This is not a problem, though, since our model is robust to partial occlusions, as shown before. Finally, the other images in Fig. 4 show some qualitative results in crowded scenes, obtained with this last classifier.

5 Conclusions

In this paper, we face three classic video surveillance applications. We propose the novel general-purpose ARCO descriptor, and we adopt a common theoretical

framework of multi-class classification on Riemannian manifold Sym_d^+. Two are the advancements. From a practical point of view ARCO can describe faces as well as pedestrians, by including arbitrary features, and exploiting their dependencies via spatially local covariances. From a theoretical point of view, we show that Sym_d^+ has non-positive sectional curvature and that where the curvature is almost flat we can perform multi-class Logitboost projecting the ARCO features on the tangent plane at any point of Sym_d^+. The experimental section validates the proposed approach, with novel state-of-the-art performances.

Acknowledgments

This research is founded by the EU-Project FP7 SAMU- RAI, grant FP7-SEC-2007-01 No. 217899.

References

1. Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation in computer vision: A survey. IEEE Trans. PAMI 31, 607–626 (2009)
2. Enzweiler, M., Gavrila, D.M.: Monocular pedestrian detection: Survey and experiments. IEEE Trans. PAMI 31, 2179–2195 (2009)
3. Yang, M., Kriegman, D., Ahuja, N.: Detecting faces in images: A survey. IEEE Trans. PAMI 24, 34–58 (2002)
4. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple. In: Proc. CVPR (2001)
5. Li, S., Zhu, L., Zhang, Z., Blake, A., Zhang, H., Shum, H.: Statistical learning of multi-view face detection. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 67–81. Springer, Heidelberg (2002)
6. Viola, M., Jones, M.J., Viola, P.: Fast multi-view face detection. In: Proc. CVPR (2003)
7. Huang, C., Ai, H., Li, Y., Lao, S.: Vector boosting for rotation invariant multi-view face detection. In: Proc. ICCV, pp. 446–453 (2005)
8. Wu, B., Ai, H., Huang, C., Lao, S.: Fast rotation invariant multi-view face detection based on real adaboost. In: FGR, pp. 79–84 (2004)
9. Li, S., Zhang, Z.: Floatboost learning and statistical face detection. IEEE Trans. PAMI 26 (2004)
10. Bar-Hillel, A., Hertz, T., Weinshall, D.: Object class recognition by boosting a part-based model. In: Proc. CVPR, pp. 702–709 (2005)
11. Tuzel, O., Porikli, F., Meer, P.: Pedestrian detection via classification on riemannian manifolds. IEEE Trans. PAMI, 1713–1727 (2008)
12. Yao, J., Odobez, J.: Fast Human Detection from Videos Using Covariance Features. In: The Eighth International Workshop on Visual Surveillance (2008)
13. Wu, B., Nevatia, R.: Optimizing discrimination-efficiency tradeoff in integrating heterogeneous local features for object detection. In: Proc. CVPR (2008)
14. Paisitkriangkrai, S., Shen, C., Zhang, J.: Performance evaluation of local features in human classification and detection. IET-CV 2, 236–246 (2008)
15. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55, 119–139 (1997)

16. Schapire, R., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Mach. Learn. 37, 297–336 (1999)
17. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: A statistical view of boosting. The Annals of Statistics 28, 337–374 (2000)
18. Wu, B., Nevatia, R.: Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. IJCV 82 (2009)
19. Tuzel, O., Porikli, F., Meer, P.: Region covariance: A fast descriptor for detection and classification. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 589–600. Springer, Heidelberg (2006)
20. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. CVPR, vol. 1, p. 886 (2005)
21. Zhang, J., Zhou, S., McMillan, L., Comaniciu, D.: Joint real-time object detection and pose estimation using probabilistic boosting network. In: Proc. CVPR, vol. 8 (2007)
22. Wu, B., Nevatia, R.: Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. In: Sanderson, J.G. (ed.) A Relational Theory of Computing. LNCS, vol. 82, pp. 185–204. Springer, Heidelberg (1980)
23. Chen, Y.-T., Chen, C.-S., Hung, Y.-P., Chang, K.-Y.: Multi-Class Multi-Instance Boosting for Part-Based Human Detection. In: ICCV 2009 Workshops, pp. 1177–1184 (2009)
24. Karcher, H.: Riemannian Center of Mass and Mollifier Smoothing. Comm. Pure and Applied Math. 30, 509–541 (1997)
25. Chavel, I.: Riemannian Geometry - A modern introduction. Cambridge University Press, Cambridge (2006)
26. Pennec, X.: Probabilities and statistics on Riemannian manifolds: a geometric approach. Technical report, INRIA (2004)
27. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Fast and simple calculus on tensors in the Log-Euclidean framework. In: Duncan, J.S., Gerig, G. (eds.) MICCAI 2005. LNCS, vol. 3749, pp. 115–122. Springer, Heidelberg (2005)
28. Breiman, L., Friedman, J., Olshen, R., Stone, C., Breiman, L., Hoeffding, W., Serfling, R., Friedman, J., Hall, O., Buhlmann, P., et al: Classification and Regression Trees. Ann. Math. Statist. 19, 293–325
29. Orozco, J., Gong, S., Xiang, T.: Head pose classification in Crowded Scenes. In: Proc. BMVC (2009)
30. Schwartz, W., Kembhavi, A., Harwood, D., Davis, L.: Human Detection Using Partial Least Squares Analysis. In: Proc. ICCV (2009)
31. Maji, S., Berg, A., Malik, J.: Classification using intersection kernel support vector machines is efficient. In: Proc. CVPR, vol. 1, p. 4 (2008)
32. Lin, Z., Davis, L.: A pose-invariant descriptor for human detection and segmentation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 423–436. Springer, Heidelberg (2008)
33. Wu, B., Nevatia, R.: Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In: ICCV (2005)
34. 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)
35. Li, M., Zhang, Z., Huang, K., Tan, T.: Estimating the number of people in crowded scenes by MID based foreground segmentation and head-shoulder detection. In: Proc. ICPR, pp. 1–4 (2008)

Modeling Temporal Structure of Decomposable Motion Segments for Activity Classification

Juan Carlos Niebles[1,2,3], Chih-Wei Chen[1], and Li Fei-Fei[1]

[1] Stanford University, Stanford CA 94305, USA
[2] Princeton University, Princeton NJ 08544, USA
[3] Universidad del Norte, Barranquilla, Colombia

Abstract. Much recent research in human activity recognition has focused on the problem of recognizing simple repetitive (walking, running, waving) and punctual actions (sitting up, opening a door, hugging). However, many interesting human activities are characterized by a complex temporal composition of simple actions. Automatic recognition of such complex actions can benefit from a good understanding of the temporal structures. We present in this paper a framework for modeling motion by exploiting the temporal structure of the human activities. In our framework, we represent activities as temporal compositions of motion segments. We train a discriminative model that encodes a temporal decomposition of video sequences, and appearance models for each motion segment. In recognition, a query video is matched to the model according to the learned appearances and motion segment decomposition. Classification is made based on the quality of matching between the motion segment classifiers and the temporal segments in the query sequence. To validate our approach, we introduce a new dataset of complex Olympic Sports activities. We show that our algorithm performs better than other state of the art methods.

Keywords: Activity recognition, discriminative classifiers.

1 Introduction

We argue that to understand motion, it is critical to incorporate temporal context information, particularly the temporal ordering of the movements. In this paper, we propose a simple discriminative framework for classifying human activities by aggregating information from motion segments that are considered both for their visual features as well as their temporal composition. An input video is automatically decomposed temporally into motion segments of variable lengths. The classifier selects a discriminative decomposition and combination of the segments for matching. Though simple in its form, we highlight a couple of advantages of our framework compared to the previous work.

First, depending on the time scale of the movement, actions have been traditionally grouped into: short but punctual actions (e.g. drink, hug), simple but periodic actions (e.g. walking, boxing), and more complex activities that are

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 392–405, 2010.

considered as a composition of shorter or simpler actions (e.g. a long jump, cooking). Very different algorithms have been proposed for these different types of motion, most of them take advantage of the special properties within its domain, hence perform rather poorly on other types. Our framework is a general one. No matter how simple or complex the motion is, our classifier relies on a temporal composition of various motion segments. Our basic philosophy is clear: temporal information helps action recognition at all time scales.

On the other hand, we note that some work has taken the approach of decomposing actions into "hidden states" that correspond to meaningful motion segments (i.e. HMM's, HCRF's, etc.). In contrast, we let the model automatically discover a robust combination of motion segments that improve the discriminability of the classifier. The result is a much simpler model that does not unnecessarily suffer from the difficult intermediate recognition step.

In order to test the efficacy of our method, we introduce a new dataset that focuses on complex motions in Olympic Sports, which can be difficult to discriminate without modeling the temporal structures. Our algorithm shows very promising results.

The rest of the paper is organized as follows. Section 1.1 overviews some of the related work. Section 2 describes a video representation that can be employed in conjunction with our model. Section 3 presents our model for capturing temporal structures in the data. We present experimental validation in Section 4 and conclude the paper in Section 5.

1.1 Related Work

A considerable amount of work has studied the recognition of human actions in video. Here we overview a few related work but refer the reader to [1,2] for a more complete survey.

A number of approaches have adopted the bag of spatio-temporal interest points [3] representation for human action recognition. This representation can be combined with either discriminative [4,5] classifiers, semi-latent topic models [6] or unsupervised generative [7,8] models. Such holistic representation of video sequences ignores temporal ordering and arrangement of features in the sequence.

Some researchers have studied the use of temporal structures for recognizing human activities. Methods based on dynamical Bayesian networks and Markov models have shown promise but either require manual design by experts [9] or detailed training data that can be expensive to collect [10]. Other work has aimed at constructing plausible temporal structures [11] in the actions of different agents but does not consider the temporal composition within the movements of a single subject, in part due to their holistic representation. On the other hand, discriminative models of temporal context have also being applied for classification of simple motions in rather simplified environments [12,13,14,15].

In addition to temporal structures, other contextual information can benefit activity recognition, such as background scene context [4] and object interactions [11,16]. Our paper focuses on incorporating temporal context, but does not exclude future work for combining more contextual information.

Our approach to capturing temporal structures is related to part-based models for object recognition. Both generative [17,18,19,20] and discriminative [21,22] models have shown promise in leveraging the spatial structures among parts for object recognition.

In this paper, we present a new representation for human activities in video. The key observation is that many activities can be described as a temporal composition of simple motion segments. At the global temporal level, we model the distinctive overall statistics of the activity. At shorter temporal ranges, we model the patterns in motion segments of shorter duration that are arranged temporally to compose the overall activity. Moreover, such temporal arrangement considered by our model is not rigid, instead it accounts for the uncertainty in the exact temporal location of each motion segment.

2 Video Representation

Our model of human actions can be applied over a variety of video descriptors. The key requirement is that a descriptor can be computed over multiple temporal scales, since our motion segment classifiers can operate on video segments of varying length. Frame-based representations and representations based on histograms are particular examples of descriptors that fit well to our framework. Here, we adopt a representation based on spatio-temporal interest points. Interest point based descriptors are attractive specially when tracking the subject performing the activity is difficult or not available. Several methods have been proposed for detecting spatio-temporal interest points in sequences [3,23,24]. In our approach, we use the 3-D Harris corner detector [3]. Each interest point is described by HoG (Histogram of Gradients) and HoF (Histogram of Flow) descriptors [5]. Furthermore, we vector quantize the descriptors by computing memberships with respect to a descriptor codebook, which is obtained by k-means clustering of the descriptors in the training set. During model learning and matching, we compute histograms of codebook memberships over particular temporal ranges of a given video, which are denoted by ψ_i in the following.

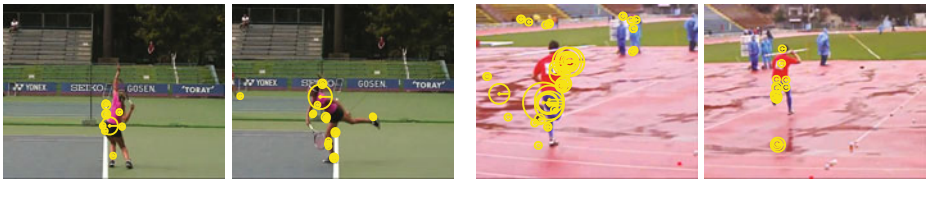


Fig. 1. Our framework can be applied over a variety of video data representations. Here we adopt a representation based on spatio-temporal interest points. This figure shows example spatio-temporal interest points detected with the 3D Harris corner method from [3]. Video patches are extracted around each point, and described by their local shape and motion patterns.

3 Modeling Temporal Structures

In this section we present our framework for recognizing complex human activities in video. We propose a temporal model for recognizing human actions that incorporates simple motion segment classifiers of multiple temporal scales. Fig. 2 shows a schematic illustration of our human action model. The basic philosophy is very simple: a video sequence is first decomposed into many temporal segments of variable length (including the degenerate case of the full sequence itself). Each video segment is matched against one of the motion segment classifiers by measuring image-based similarities as well as the temporal location of the segment with respect to the full sequence. The best matching scores from each motion segment classifier are accumulated to obtain a measure of the matching quality between the full action model and the query video. As Fig. 2 illustrates, an action model encodes motion information at multiple temporal scales. It also encodes the ordering in which the motion segments tend to appear in the sequence. In the following, we discuss the details of the model, the recognition process and learning algorithm.

3.1 Model Description

Here we introduce the model of human actions, which is illustrated in Fig. 2. Our full action model is composed by a set of K motion segment classifiers $A_1, ..., A_K$, each of them operating at a particular temporal scale. Each motion segment classifier A_i operates over a histogram of quantized interest points extracted from a temporal segment whose length is defined by the classifier's temporal scale s_i. In addition to the temporal scale, each motion segment classifier also specifies a temporal location centered at its preferred anchor point t_i. Lastly, the motion segment classifier is enriched with a flexible displacement model τ_i that captures the variability in the exact placement of the motion segment A_i within the sequence.

We summarize the parameters of our model with the parameter vector $\mathbf{w}$ as the concatenation of the motion segment classifiers and the temporal displacement parameters,

$$\mathbf{w} = (A_1, ..., A_K, \tau_1, ..., \tau_K). \tag{1}$$

3.2 Model Properties

Our model addresses the need to consider temporal structure in the task of human activity classification. In the following, we discuss some important properties of our framework.

Coarse-to-fine motion segment classifiers. Our model contains multiple classifiers at different time scales, enabling it to capture characteristic motions of various temporal granularity. On one end, holistic bag-of-features operate at the coarsest scale, while frame-based methods operate at the finest scale. Our

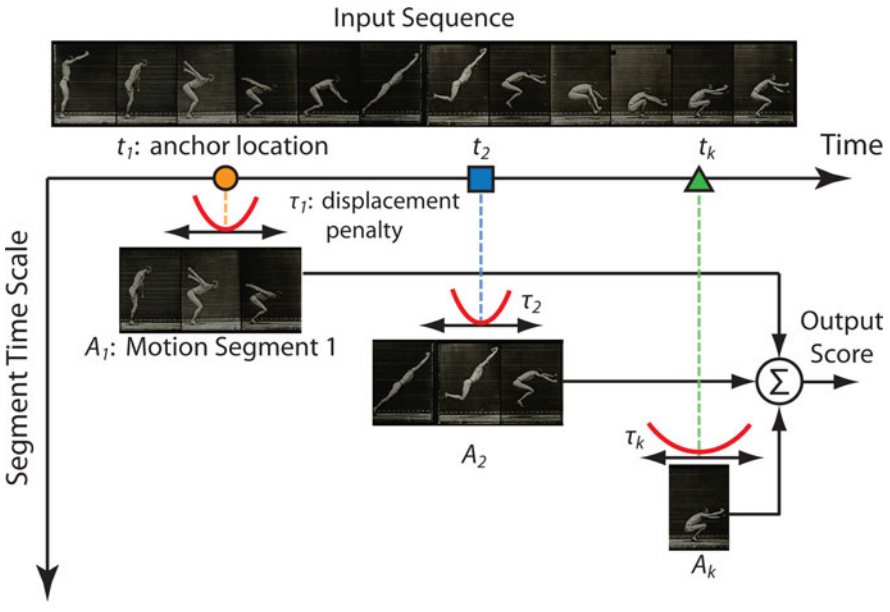


Fig. 2. Model Architecture. Here we show the structure of our model for activity recognition. The input video V is described by histograms of vector quantized interest points, which are computed over multiple temporal ranges. Each motion segment classifier A_i has a particular temporal scale, and it is matched to the features $\psi_i(V, h_i)$ from temporal segments of the input sequence of that temporal extent. The optimal location of each motion segment classifier is determined by the appearance similarity $(A_i \cdot \psi_i(V, h_i))$ and penalty of temporal displacement from the anchor point t_i $(\tau_i \cdot \psi_(h_i - t_i))$. The overall matching score combines scores of individual components. A classification decision is made by thresholding the resulting matching score. See Sec. 3 for more details.

framework has the flexibility to operate between these two ends of the temporal spectrum, and it closes the gap by allowing multiple classifiers to reside in a continuum of temporal scales.

Temporal Context. While discriminative appearance is captured by our multiple classifiers at different time scales, the location and order in which the motion segments occur in the overall activity also offer rich information about the activity itself. Our framework is able to capture such temporal context: the anchor points of the motion segment classifiers encode the temporal structure of the activity. In particular, these canonical positions prohibit the classifiers from matching time segments that are distant from them. This implicitly carries ordering constraints that are useful for discriminating human activities.

Flexible Model. Equipped with classifiers of multiple time scales and the temporal structure embedded in their anchor points, our model is capable of searching for a best match in a query sequence and score it accordingly. However, the

temporal structure in videos of the same class might not be perfectly aligned. To handle intra-class variance, our model incorporates a temporal displacement penalty that allows the optimal placement of the each motion segment to deviate from its anchor point.

3.3 Recognition

Given a trained model, the task in recognition is to find the best matching of the model to an input sequence. This requires finding the best scoring placement for each of the K motion segment classifiers. We denote a particular placement of the motion segment classifiers within a sequence V by a hypothesis $H = (h_1, ..., h_k)$. Each h_i defines the temporal position for the i-th motion segment classifier. We measure the matching quality of motion segment classifier A_i at location h_i by favoring good appearance similarity between the motion segment classifier and the video features, and penalizing for the temporal misplacement of the motion segment classifier when h_i is far from the anchor point t_i. That is, the matching score for the i-th motion segment classifier is

$$A_i \cdot \psi_i(V, h_i) - \tau_i \cdot \psi_{di}(h_i - t_i). \tag{2}$$

In the first term of Eq. 2, which captures the appearance similarity, $\psi_i(V, h_i)$ is the appearance feature vector (i.e. histogram of quantized interest points) extracted at location h_i with scale s_i. In our experiments, we implement the classifier A_i with a χ^2 support vector machine. The kernel function for A_i is given by

$$K(x_k, x_j) = \exp\left(-\frac{1}{2S}\sum_{r=1}^{D}\frac{(x_{kr} - x_{jr})^2}{x_{kr} + x_{jr}}\right), \tag{3}$$

where S denotes the mean distance among training examples, $\{x_{ki}\}_{i=1...D}$ are the elements of the histogram x_k and D is the histogram dimensionality. In practice, D is equal to the size of the codebook. In the second term of Eq. 2, which captures the temporal misplacement penalty, $\psi_{di}(h_i - t_i)$ denotes the displacement feature. The penalty, parametrized by $\tau_i = \{\alpha_i, \beta_i\}$, is a quadratic function of the motion segment displacement and given by

$$\tau_i \cdot \psi_{di}(h_i - t_i) = \alpha_i \cdot (h_i - t_i)^2 + \beta_i \cdot (h_i - t_i). \tag{4}$$

We obtain an overall matching score for hypothesis H by accumulating the scores from all motion segment classifiers in the model:

$$\sum_{i=1}^{K} A_i \cdot \psi_i(V, h_i) - \tau_i \cdot \psi_{di}(h_i - t_i). \tag{5}$$

Let $f_{\mathbf{W}}(V)$ be a scoring function that evaluates sequence V. In recognition, we consider all possible hypotheses and choose the one with the best matching score:

$$f_{\mathbf{w}}(V) = \max_{H} \sum_{i=1}^{K} A_i \cdot \psi_i(V, h_i) - \tau_i \cdot \psi_{di}(h_i - t_i). \quad (6)$$

A binary classification decision for input video V is done by thresholding the matching score $f_{\mathbf{w}}(V)$.

There is a large number of hypotheses for a given input video sequence. However, note that once the appearance similarities between the video sequence and each motion segment classifier are computed, selecting the hypothesis with the best matching score can be done efficiently using dynamic programming and distance transform techniques [18] in a similar fashion to [21,25].

3.4 Learning

Suppose we are given a set of example sequences $\{V^1, \ldots, V^N\}$ and their corresponding class labels $y_{1:N}$, with $y_i \in \{1, -1\}$. Our goal is to use the training examples to learn the model parameters $\mathbf{w}$. This can be formulated as the minimization of a discriminative cost function. In particular, we consider the following minimization problem:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{N} \max(0, 1 - y_i f_{\mathbf{w}}(V^i)), \quad (7)$$

where C controls the relative weight of the hinge loss term. This is the formulation of a Latent Support Vector Machine (LSVM) [21]. In the LSVM framework, the scoring function maximizes over the hidden variables. In our method, the hidden variables correspond to the best locations of the motion segment classifiers on each training video. Note that it is not necessary to supervise the locations of the motion segment classifiers during training, instead this is a weakly supervised setting, where only a class label is provided for each example.

The optimization problem described above is, in general, non-convex. However, it has been shown in [21] that the objective function is convex for the negative examples, and also convex for the positive examples when the hidden variables are fixed.

This leads to an iterative learning algorithm that alternates between estimating model parameters and estimating the hidden variables for the positive training examples. In summary the procedure is as follows. In the first step, the model parameters $\mathbf{w}$ are fixed. The best scoring locations $H_p^\star$ of the motion segment classifiers are selected for each positive example p. This is achieved by running the matching process described in Section 3.3 on the positive videos. In the second step, by fixing the hidden variables of the positive examples to the locations given by $H_p^\star$, the optimization problem in Eq. 7 becomes convex. We select negative examples by running the matching process in all negative training videos and retrieving all hypotheses with large matching score. We train the parameters $\mathbf{w}$ using LIBSVM [26] on the resulting positive and negative examples. This process is repeated for a fixed small number of iterations.

In most cases, the iterative algorithm described above requires careful initialization. We choose a simple initialization heuristic. First, we train a classifier

Table 1. Left: Accuracy for action classification in the KTH dataset. Right: Comparison of our model to current state of the art methods.

Action Class	Our Model
walking	94.4%
running	79.5%
jogging	78.2%
hand-waving	99.9%
hand-clapping	96.5%
boxing	99.2%

Algorithm	Perf.
Ours	91.3%
Wang et al. [28]	92.1%
Laptev et al. [5]	91.8%
Wong et al. [8]	86.7%
Schuldt et al. [27]	71.5%
Kim et al. [29]	95%

with a single motion segment classifier that covers the entire sequence. This is equivalent to training a χ^2-SVM on a holistic bag of features representation. We then augment the model with the remaining $K-1$ motion segment classifiers. The location and scale of each additional motion segment classifier is selected so that it covers a temporal range that correlates well with the global motion segment classifier. This favors temporal segments that exhibit features important for overall discrimination.

4 Experimental Results

In order to test our framework, we consider three experimental scenarios. First, we test the ability of our approach to discriminate simple actions on a benchmark dataset. Second, we test the effectiveness of our model at leveraging the temporal structure in human actions on a set of synthesized complex actions. Last, we present a new challenging Olympic Sports Dataset and show promising classification results with our method.

4.1 Simple Actions

We use the KTH Human actions dataset [27] to test the ability of our method to classify simple motions. The dataset contains 6 actions performed by 25 actors, for a total of 2396 sequences. We follow the experimental settings described in [27]. In all experiments, we adopt a representation based on spatio-temporal interest points described by concatenated HoG/HoF descriptors. We construct a codebook of local spatio-temporal patches from feature descriptors in the training set. We set the number of codewords to be 1000. Experimental results are shown in Table 1. A direct comparison is possible to the methods that follow the same experimental setup [5,8,27,28]. We note that our method shows competitive results, but its classification accuracy is slightly lower than the best result reported in [28].

4.2 Synthesized Complex Actions

In this experiment, we aim to test the ability of our model to leverage the temporal structure of human actions. In order to test this property in a controlled

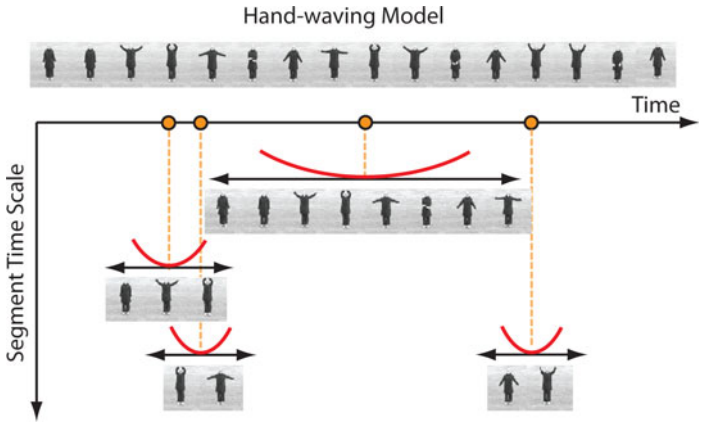


Fig. 3. An example of our learned model. In this illustration, the horizontal axis represents time. Each row corresponds to a motion segment classifier learned by our model whose temporal extent is indicated by its vertical location. The appearance of the motion segment is illustrated by a few example frames. The associated dot indicates the anchor position t_i of the motion segment relative to the full sequence. The parameters of the temporal misplacement penalty τ_i are represented by the parabola centered at the anchor point. Notice that the vertical arrangement of the motion segments shows the distinct temporal scales at which each classifier operates.

setting, we construct a synthesized set of complex actions by concatenating 3 simple motions from the Weizmann action database: 'jump', 'wave' and 'jack'. In total, we synthesize 6 complex actions classes by concatenating one video of each simple motion into a long sequence.

In this test, a baseline model that uses a single motion segment classifier covering the entire video sequence performs at random chance or $\approx 17\%$. The simple holistic bag-of-features has trouble differentiating actions in this set since the overall statistics are nearly identical. On the other hand, our model which takes advantage of temporal structure and orderings, can easily discriminate the 6 classes and achieve perfect classification performance at 100%. In Fig. 4 we show a learned model for the complex action composed by 'wave'-'jump'-'jack'. Notice that our model nicely captures discriminative motion segments such as the transitions between 'jump' and 'jack'.

4.3 Complex Activities: Olympic Sports Dataset

We have collected a dataset of Olympic Sports activities from YouTube sequences. Our dataset contains 16 sport classes, with 50 sequences per class. See Fig. 5 for example frames from the dataset. The sport activities depicted in the dataset contain complex motions that go beyond simple punctual or repetitive actions[1]. For instance, sequences from the long-jump action class, show an

[1] In contrast to other sport datasets such as [15], which contains periodic or simple actions such as walking, running, golf-swing, ball-kicking.

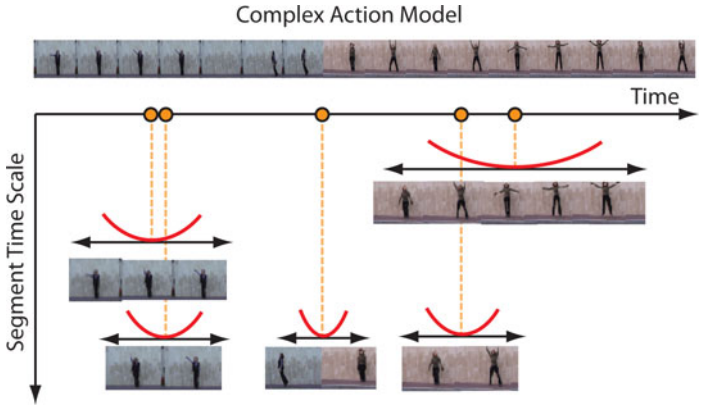


Fig. 4. A learned model for the synthesized complex action 'wave'-'jump'-'jack'. See Fig. 3 for a description of the illustration.

Fig. 5. Olympic Sports Dataset. Our dataset contains 50 videos from each of 16 classes: high jump, long jump, triple jump, pole vault, discus throw, hammer throw, javelin throw, shot put, basketball layup, bowling, tennis serve, platform (diving), springboard (diving), snatch (weightlifting), clean and jerk (weightlifting) and vault (gymnastics). The sequences, obtained from YouTube, contain severe occlusions, camera movements, compression artifacts, etc. The dataset is available at `http://vision.stanford.edu`.

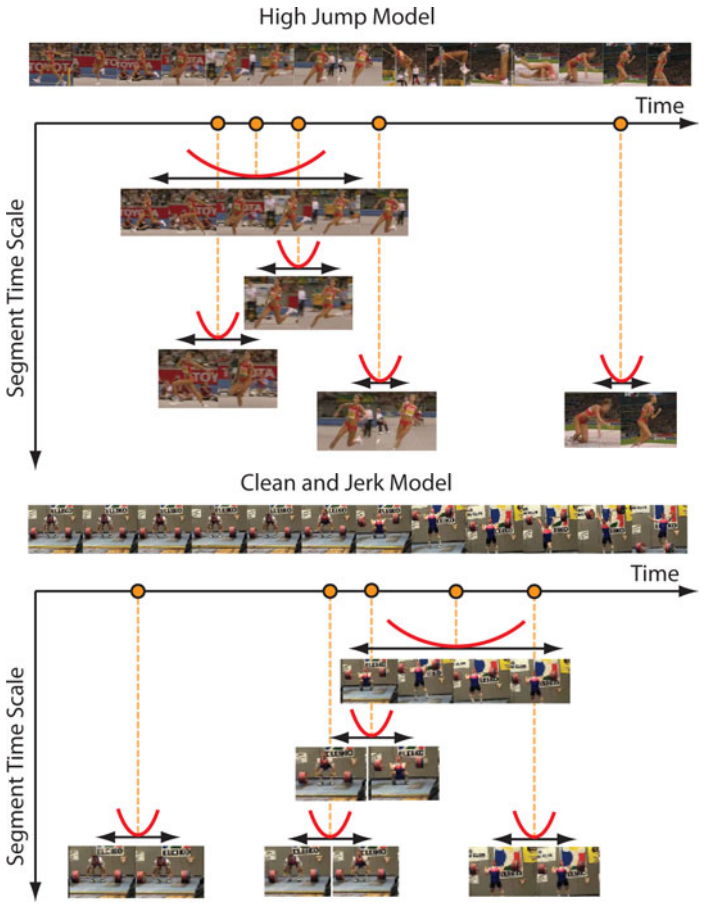


Fig. 6. Learned model for the complex actions in the Olympic Sports Dataset: high-jump and clean-and-jerk. See Fig. 3 for a description of the illustration.

athlete first standing still, in preparation for his/her jump, followed by running, jumping, landing and finally standing up. The dataset is available for download at our website `http://vision.stanford.edu`.

We split the videos from each class in the dataset into 40 sequences for training and 10 for testing. We illustrate two of the learned models in Fig. 6. Table 2 shows the classification results of our algorithm. We compare the performance of our model to the multi-channel method of [5], which incorporates rigid spatio-temporal binnings and captures a rough temporal ordering of features.

Finally, Fig. 7 shows three learned models of actions in the Olympic Sports dataset, along with matchings to some testing sequences. In the long jump example, the first motion segment classifier covers the running motion at the beginning of the sequence. This motion segment has a low displacement penalty over a large temporal range as indicated by its wide parabola. It suggests that the model has

Fig. 7. We illustrate learned action models for long jump, vault and snatch. Each group depicts two testing sequences (top and bottom), as well as an illustration of the temporal displacement penalty parameters (middle). Green boxes surround matched temporal segments that are most compatible with the corresponding motion segment classifiers. Red boxes indicate temporal segments that are matched to the motion segment model with a low matching score. The arrows indicate the automatically selected best placement for each motion segment.

Table 2. Average Precision (AP) values for the classification task in our Olympic Sports Dataset

Sport class	Our Method	Laptev et al. [5]	Sport class	Our Method	Laptev et al. [5]
high-jump	**68.9%**	52.4%	javelin-throw	**74.6%**	61.1%
long-jump	**74.8%**	66.8%	hammer-throw	**77.5%**	65.1%
triple-jump	**52.3%**	36.1%	discus-throw	**58.5%**	37.4%
pole-vault	**82.0%**	47.8%	diving-platform	87.2%	**91.5%**
gymnastics-vault	86.1%	**88.6%**	diving-springboard	77.2%	**80.7%**
shot-put	**62.1%**	56.2%	basketball-layup	**77.9%**	75.8%
snatch	**69.2%**	41.8%	bowling	**72.7%**	66.7%
clean-jerk	**84.1%**	83.2%	tennis-serve	**49.1%**	39.6%
			Average (AAP)	**72.1%**	62.0%

learned to tolerate large displacements in the running stage of this activity. On the other hand, in the vault example, the middle motion segment classifier has a low matching score to the top testing sequence. However, the matching scores in other temporal segments are high, which provides enough evidence to the full action model for classifying this sequence correctly. Similarly, the bottom clean and jerk sequence in the snatch model obtains a high matching score for the last motion segment, but the evidence from the motion segments is rather low. We also observe that our learned motion segment classifiers display a wide range of temporal scales, indicating that our model is able to capture characteristic motion patterns at multiple scales. For example, the longer segments that contain the athlete holding the weights in the snatch model, and the shorter segments that enclose a jumping person in the long jump model.

5 Conclusion and Future Work

In this paper we have empirically shown that incorporating temporal structures is beneficial for recognizing both complex human activities as well as simple actions. Future directions include incorporating other types of contextual information and richer video representations.

Acknowledgments. This project is partially supported by an NSF CAREER grant and a Kodak Award to L.F-F. We would like to thank Barry Chai, Olga Russakovsky, Hao Su, Bangpeng Yao and the anonymous reviewers for their useful comments.

References

1. Turaga, P., Chellappa, R., Subrahmanian, V.S., Udrea, O.: Machine Recognition of Human Activities: A Survey. IEEE Transactions on Circuits and Systems for Video Technology 18, 1473–1488 (2008)
2. Forsyth, D.A., Arikan, O., Ikemoto, L., O'Brien, J., Ramanan, D.: Computational Studies of Human Motion: Part 1, Tracking and Motion Synthesis. Foundations and Trends in Computer Graphics and Vision 1, 77–254 (2005)
3. Laptev, I.: On Space-Time Interest Points. IJCV 64, 107–123 (2005)
4. Marszalek, M., Laptev, I., Schmid, C.: Actions in context. In: CVPR, pp. 2929–2936. IEEE, Los Alamitos (2009)
5. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR, p. 18. IEEE, Los Alamitos (2008)
6. Wang, Y., Mori, G.: Human action recognition by semilatent topic models. IEEE TPAMI 31, 1762–1774 (2009)
7. Niebles, J.C., Wang, H., Fei-Fei, L.: Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. IJCV 79, 299–318 (2008)
8. Wong, S.F., Kim, T.K., Cipolla, R.: Learning Motion Categories using both Semantic and Structural Information. In: CVPR, pp. 1–6. IEEE, Los Alamitos (2007)
9. Laxton, B., Lim, J., Kriegman, D.: Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video. In: CVPR. IEEE, Los Alamitos (2007)

10. Ikizler, N., Forsyth, D.A.: Searching for Complex Human Activities with No Visual Examples. IJCV 80, 337–357 (2008)
11. Gupta, A., Srinivasan, P., Shi, J., Davis, L.S.: Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In: CVPR, pp. 2012–2019. IEEE, Los Alamitos (2009)
12. Sminchisescu, C., Kanaujia, A., Metaxas, D.: Conditional models for contextual human motion recognition. CVIU 104, 210–220 (2006)
13. Wang, S.B., Quattoni, A., Morency, L.P., Demirdjian, D., Darrell, T.: Hidden Conditional Random Fields for Gesture Recognition. In: CVPR, vol. 2, pp. 1521–1527. IEEE, Los Alamitos (2006)
14. Quattoni, A., Wang, S.B., Morency, L.P., Collins, M., Darrell, T.: Hidden conditional random fields. IEEE TPAMI 29, 1848–1853 (2007)
15. Rodriguez, M.D., Ahmed, J., Shah, M.: Action MACH a spatio-temporal Maximum Average Correlation Height filter for action recognition. In: CVPR. IEEE, Los Alamitos (2008)
16. Yao, B., Fei-Fei, L.: Modeling Mutual Context of Object and Human Pose in Human-Object Interaction Activities. In: CVPR. IEEE, Los Alamitos (2010)
17. Bouchard, G., Triggs, B.: Hierarchical Part-Based Visual Object Categorization. In: CVPR, pp. 710–715. IEEE, Los Alamitos (2005)
18. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial Structures for Object Recognition. IJCV 61, 55–79 (2005)
19. Fergus, R., Perona, P., Zisserman, A.: Weakly Supervised Scale-Invariant Learning of Models for Visual Recognition. IJCV 71, 273–303 (2007)
20. Niebles, J.C., Fei-Fei, L.: A Hierarchical Model of Shape and Appearance for Human Action Classification. In: CVPR, pp. 1–8. IEEE, Los Alamitos (2007)
21. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object Detection with Discriminatively Trained Part Based Models. IEEE TPAMI, 1–20 (2009)
22. Ke, Y., Sukthankar, R., Hebert, M.: Event Detection in Crowded Videos. In: ICCV, pp. 1–8. IEEE, Los Alamitos (2007)
23. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior Recognition via Sparse Spatio-Temporal Features. In: VSPETS, pp. 65–72. IEEE, Los Alamitos (2005)
24. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as Space-Time Shapes. In: ICCV, vol. 2, pp. 1395–1402. IEEE, Los Alamitos (2005)
25. Felzenszwalb, P.F., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR, pp. 1–8. IEEE, Los Alamitos (2008)
26. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm
27. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: ICPR, pp. 32–36. IEEE, Los Alamitos (2004)
28. Wang, H., Ullah, M.M., Klaser, A., Laptev, I., Schmid, C.: Evaluation of local spatio-temporal features for action recognition. In: BMVC (2009)
29. Kim, T.K., Wong, S.F., Cipolla, R.: Tensor Canonical Correlation Analysis for Action Classification. In: CVPR, pp. 1–8. IEEE, Los Alamitos (2007)

Cascaded Models for Articulated Pose Estimation

Benjamin Sapp, Alexander Toshev, and Ben Taskar

University of Pennsylvania,
Philadelphia, PA 19104 USA
{bensapp,toshev,taskar}@cis.upenn.edu

Abstract. We address the problem of articulated human pose estimation by learning a coarse-to-fine cascade of pictorial structure models. While the fine-level state-space of poses of individual parts is too large to permit the use of rich appearance models, most possibilities can be ruled out by efficient structured models at a coarser scale. We propose to learn a sequence of structured models at different pose resolutions, where coarse models filter the pose space for the next level via their max-marginals. The cascade is trained to prune as much as possible while preserving true poses for the final level pictorial structure model. The final level uses much more expensive segmentation, contour and shape features in the model for the remaining filtered set of candidates. We evaluate our framework on the challenging Buffy and PASCAL human pose datasets, improving the state-of-the-art.

1 Introduction

Pictorial structure models [1] are a popular method for human body pose estimation [2,3,4,5,6]. The model is a Conditional Random Field over pose variables that characterizes local appearance properties of parts and geometric part-part interactions. The search over the joint pose space is linear time in the number of parts when the part-part dependencies form a tree. However, the individual part state-spaces are too large (typically hundreds of thousands of states) to allow complex appearance models be evaluated densely. Most appearance models are therefore simple linear filters on edges, color and location [2,4,5,6]. Similarly, because of quadratic state-space complexity, part-part relationships are typically restricted to be image-independent deformation costs that allow for convolution or distance transform tricks to speed up inference [2]. A common problem in such models is poor localization of parts that have weak appearance cues or are easily confused with background clutter (accuracy for lower arms in human figures is almost half of that for torso or head [6]). Localizing these elusive parts requires richer models of individual part shape and joint part-part appearance, including contour continuation and segmentation cues, which are prohibitive to compute densely.

In order to enable richer appearance models, we propose to learn a cascade of pictorial structures (CPS) of increasing pose resolution which progressively

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 406–420, 2010.

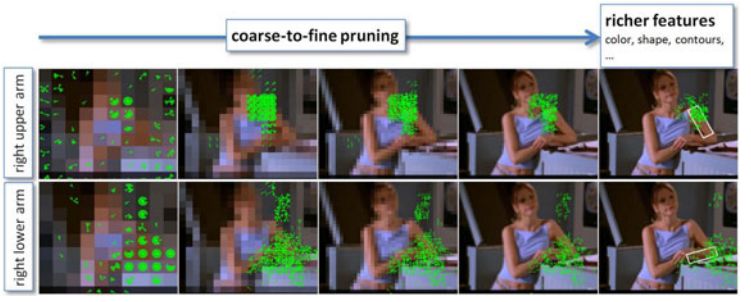


Fig. 1. Overview: A discriminative coarse-to-fine cascade of pictorial structures filters the pose space so that expressive and computationally expensive cues can be used in the final pictorial structure. Shown are 5 levels of our coarse-to-fine cascade for the right upper and lower arm parts. Green vectors represent position and angle of unpruned states, the downsampled images correspond to the dimensions of the resepective state space, and the white rectangles represent classification using our final model.

filter the pose state space. Conceptually, the idea is similar to the work on cascades for face detection [7,8], but the key difference is the use of structured models. Each level of the cascade at a given spatial/angular resolution refines the set of candidates from the previous level and then runs inference to determine which poses to filter out. For each part, the model selects poses with the largest max-marginal scores, subject to a computational budget. Unlike conventional pruning heuristics, where the possible part locations are identified using the output of a detector, models in our cascade use inference in simpler structured models to identify what to prune, taking into account global pose in filtering decisions. As a result, at the final level the CPS model has to deal with a much smaller hypotheses set which allows us to use a rich combination of features. In addition to the traditional part detectors and geometric features, we are able to incorporate object boundary continuity and smoothness, as well as shape features. The former features represent mid-level and bottom-up cues, while the latter capture shape information, which is complementary to the traditional HoG-based part models. The approach is illustrated in the overview Figure 1. We apply the presented CPS model combined with the richer set of features on the Buffy and PASCAL stickmen benchmark, improving the state-of-the-art on arm localization.

2 Related Work

The literature on human pose estimation is vast and varied in settings: applications range from highly-constrained MOCAP environments (*e.g.* [9]) to extremely articulated baseball players (*e.g.* [10]) to the recently popular "in the wild" datasets Buffy (from TV) and the PASCAL Stickmen (from amateur photographs) [5]. We focus our attention here on the work most similar in spirit to

ours, namely, pictorial structures models. First proposed in [1], efficient inference methods focusing on tree-based models with quadratic deformation costs were introduced in [2]. Ramanan [4] proposed learning PS parameters discriminitively by maximizing conditional likelihood and introduced further improvements using iterative EM-like parsing [11]. Ferrari et al. [5,12] also prune the search space for computational efficiency and to avoid false positives. Our end goal is the same, but we adopt a more principled approach, expressing features on regions and locations and letting our system learn what to eliminate at run-time given the image.

For unstructured, binary classification, cascades of classifiers have been quite successful for reducing computation. Fleuret and Geman [7] propose a coarse-to-fine sequence of binary tests to detect the presence and pose of objects in an image. The learned sequence of tests is trained to minimize expected computational cost. The extremely popular Viola-Jones classifier [8] implements a cascade of boosting ensembles, with earlier stages using fewer features to quickly reject large portions of the state space.

Our cascade model is inspired by these binary classification cascades, and is based on the structured prediction cascades framework [13]. In natural language parsing, several works [14,15] use a coarse-to-fine idea closely related to ours and [7]: the marginals of a simple context free grammar or dependency model are used to prune the parse chart for a more complex grammar.

Recently, Felzenszwalb et al. [16] proposed a cascade for a structured parts-based model. Their cascade works by early stopping while evaluating individual parts, if the combined part scores are less than fixed thresholds. While the form of this cascade can be posed in our more general framework (a cascade of models with an increasing number of parts), we differ from [16] in that our pruning is based on thresholds that adapt based on inference in each test example, and we explicitly learn parameters in order to prune safely and efficiently. In [7,8,16], the focus is on preserving established levels of accuracy while increasing speed. The focus in this paper is instead developing more complex models—previously infeasible due to the original intractable complexity—to improve state-of-the-art performance.

A different approach to reduce the intractable number of state hypotheses is to instead propose a small set of likely hypotheses based on bottom-up perceptual grouping principles [10,17]. Mori et al. [10] use bottom-up saliency cues, for example strength of supporting contours, to generate limb hypotheses. They then prune via hand-set rules based on part-pair geometry and color consistency. The shape, color and contour based features we use in our last cascade stage are inspired by such bottom-up processes. However, our cascade is solely a sequence of discriminatively-trained top-down models.

3 Framework

We first summarize the basic pictorial structure model and then describe the inference and learning in the cascaded pictorial structures.

Classical pictorial structures are a class of graphical models where the nodes of the graph represents object parts, and edges between parts encode pairwise geometric relationships. For modeling human pose, the standard PS model decomposes as a tree structure into unary potentials (also referred to as appearance terms) and pairwise terms between pairs of physically connected parts. Figure 2 shows a PS model for 6 upper body parts, with lower arms connected to upper arms, and upper arms and head connected to torso. In previous work [4,2,5,12,6], the pairwise terms do not depend on data and are hence referred to as a spatial or structural prior. The state of part L_i, denoted as $l_i \in \mathcal{L}_i$, encodes the joint location of the part in image coordinates and the direction of the limb as a unit vector: $l_i = [l_{ix}\ l_{iy}\ l_{iu}\ l_{iv}]^T$. The state of the model is the collection of states of M parts: $p(L = l) = p(L_1 = l_1, \ldots, L_M = l_M)$. The size of the state space for each part, $|\mathcal{L}_i|$, the number of possible locations in the image times the number of predefined discretized angles. For example, standard PS implementations typically model the state space of each part in a roughly 100×100 grid for $l_{ix} \times l_{iy}$, with 24 different possible values of angles, yielding $|\mathcal{L}_i| = 100 \times 100 \times 24 = 240,000$. The standard PS formulation (see [2]) is usually written in a log-quadratic form:

$$p(l|x) \propto \prod_{ij} \exp(-\frac{1}{2}||\Sigma_{ij}^{-1/2}(T_{ij}(l_i) - l_j - \mu_{ij})||_2^2) \times \prod_{i=1}^{M} \exp(\mu_i^T \phi_i(l_i, x)) \quad (1)$$

The parameters of the model are μ_i, μ_{ij} and Σ_{ij}, and $\phi_i(l_i, x)$ are features of the (image) data x at location/angle l_i. The affine mapping T_{ij} transforms the part coordinates into a relative reference frame. The PS model can be interpreted as a set of springs at rest in default positions μ_{ij}, and stretched according to tightness Σ_{ij}^{-1} and displacement $\phi_{ij}(l) = T_{ij}(l_i) - l_j$. The unary terms pull the springs toward locations l_i with higher scores $\mu_i^T \phi_i(l_i, x)$ which are more likely to be a location for part i.

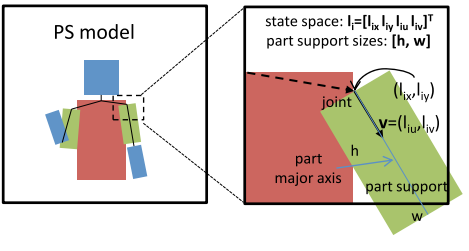


Fig. 2. Basic PS model with state l_i for a part L_i

This form of the pairwise potentials allows inference to be performed faster than $O(|\mathcal{L}_i|^2)$: MAP estimates $\arg\max_l p(l|x)$ can be computed efficiently using a generalized distance transform for max-product message passing in $O(|\mathcal{L}_i|)$ time. Marginals of the distribution, $p(l_i|x)$, can be computed efficiently using FFT convolution for sum-product message passing in $O(|\mathcal{L}_i| \log |\mathcal{L}_i|)$ [2].

While fast to compute and intuitive from a spring-model perspective, this model has two significant limitations. One, the pairwise costs are unimodal Gaussians, which cannot capture the true multimodal interactions between pairs of body parts. Two, the pairwise terms are only a function of the geometry of the state configuration, and are oblivious

to the image cues, for example, appearance similarity or contour continuity of the a pair of parts.

We choose instead to model part configurations as a general log-linear Conditional Random Field over pairwise and unary terms:

$$p(l|x) \propto \exp\left[\sum_{ij} \theta_{ij}^T \phi_{ij}(l_i, l_j, x) + \sum_i \theta_i^T \phi_i(l_i, x)\right] = e^{\theta^T \phi(l,x)}. \quad (2)$$

The parameters of our model are the pairwise and unary weight vectors θ_{ij} and θ_i corresponding to the pairwise and unary feature vectors $\phi_{ij}(l_i, l_j, x)$ and $\phi_i(l_i, x)$. For brevity, we stack all the parameters and features into vectors using notation $\theta^T \phi(l, x)$. The key differences with the classical PS model are that (1) our pairwise costs allow data-dependent terms, and (2) we do not constrain our parameters to fit any parametric distribution such as a Gaussian. For example, we can express the pairwise features used in the classical model as $l_i \cdot l_i$, $l_j \cdot l_j$ and $l_i \cdot l_j$ without requiring that their corresponding weights can be combined into a positive semi-definite covariance matrix.

In this general form, inference can not be performed efficiently with distance transforms or convolution, and we rely on standard $O(|\mathcal{L}_i|^2)$ dynamic programming techniques to compute the MAP assignment or part posteriors. Many highly-effective pairwise features one might design would be intractable to compute in this manner for a reasonably-sized state space—for example an 100×100 image with a part angle discretization of 24 bins yields $|\mathcal{L}_i|^2 = 57.6$ billion part-part hypotheses.

In the next section, we describe how we circumvent this issue via a cascade of models which aggressively prune the state space at each stage typically without discarding the correct sequence. After the state space is pruned, we are left with a small enough number of states to be able to incorporate powerful data-dependent pairwise and unary features into our model.

Structured Prediction Cascades

The recently introduced Structured Prediction Cascade framework [13] provides a principled way to prune the state space of a structured prediction problem via a sequence of increasingly complex models. There are many possible ways of defining a sequence of increasingly complex models. In [13] the authors introduce higher-order cliques into their models in successive stages (first unary, then pairwise, ternary, etc.). Another option is to start with simple but computationally efficient features, and add more complex features downstream as the number of states decreases. Yet another option is to geometrically coarsen the original state space and successively prune and refine. We use a coarse-to-fine state space approach with simple features until we are at a reasonably fine enough state space resolution and left with few enough states that we can introduce more complex features. We start with a severely coarsened state space and use standard pictorial structures unary detector scores and geometric features to perform quick exhaustive inference on the coarse state space.

More specifically, each level of the cascade uses inference to identify which states to prune away and the next level refines the spatial/angular resolution on the unpruned states. The key ingredient to the cascade framework is that states are pruned using *max-marginal* scores, computed using dynamic programming techniques. For brevity of notation, define the score of a joint part state l as $\theta_x(l)$ and the max-marginal score of a part state as follows:

$$\theta_x(l) = \theta^T \phi(l, x) = \sum_{ij} \theta_{ij}^T \phi_{ij}(l_i, l_j, x) + \sum_i \theta_i^T \phi_i(l_i, x) \tag{3}$$

$$\theta_x^\star(l_i) = \max_{l' \in L} \quad \{\theta_x(l') \ : \ l'_i = l_i\} \tag{4}$$

In words, the max-marginal for location/angle l_i is the score of the best sequence which constrains $L_i = l_i$. In a pictorial structure model, this corresponds to fixing limb i at location l_i, and determining the highest scoring configuration of other part locations and angles under this constraint. A part could have weak individual image evidence of being at location l_i but still have a high max-marginal score if the rest of the model believes this is a likely location. Similarly, we denote the MAP assignment score as $\theta_x^\star = \max_{l \in L} \theta_x(l)$, the unconstrained best configuration of all parts.

When learning a cascade, we have two competing objectives that we must trade off, accuracy and efficiency: we want to minimize the number of errors incurred by each level of the cascade and maximize the number of filtered max marginals. A natural strategy is to prune away the lowest ranked states based on max-marginal scores. Instead, [13] prune the states whose max-marginal score is lower than an data-specific threshold t_x: l_i is pruned if $\theta_x^\star(l_i) < t_x$. This threshold is defined as a convex combination of the MAP assignment score and the mean max-marginal score, meant to approximate a percentile threshold:

$$t_x(\theta, \alpha) = \alpha \theta_x^\star + (1 - \alpha) \frac{1}{M} \sum_{i=1}^{M} \frac{1}{|\mathcal{L}_i|} \sum_{l_i \in \mathcal{L}_i} \theta_x^\star(l_i),$$

where $\alpha \in [0, 1]$ is a parameter to be chosen that determines how aggressively to prune. When $\alpha = 1$, only the best state is kept, which is equivalent to finding the MAP assignment. When $\alpha = 0$ approximately half of the states are pruned (if the median of max-marginals is equal to the mean). The advantage of using $t_x(\theta, \alpha)$ is that it is convex in θ, and leads to a convex formulation for parameter estimation that trades off the proportion of incorrectly pruned states with the proportion of unpruned states. Note that α controls efficiency, so we focus on learning the parameters θ that minimize the number of errors for a given filtering level α. The learning formulation uses a simple fact about max-marginals and the definition of $t_x(\theta, \alpha)$ to get a handle on errors of the cascade: if $\theta_x(l) > t_x(\theta, \alpha)$, then for all i, $\theta_x^\star(l_i) > t_x(\theta, \alpha)$, so no part state of l is pruned. Given an example (x, l), this condition $\theta_x(l) > t_x(\theta, \alpha)$ is sufficient to ensure that no correct part is pruned.

To learn one level of the structured cascade model θ for a fixed α, we try to minimize the number of correct states that are pruned on training data by

solving the following convex margin optimization problem given N training examples (x^n, l^n):

$$\min_{\theta} \quad \frac{\lambda}{2}||\theta||^2 + \frac{1}{N}\sum_{n=1}^{N} H(\theta; x^n, l^n), \tag{5}$$

where H is a hinge upper bound $H(\theta; x, l) = \max\{0, 1 + t_x(\theta, \alpha) - \theta_x(l)\}$. The upper-bound H is a hinge loss measuring the margin between the filter threshold $t_{x^n}(\theta, \alpha)$ and the score of the truth $\theta^T\phi(l^n, x^n)$; the loss is zero if the truth scores above the threshold by margin 1. We solve (5) using stochastic sub-gradient descent. Given an example (x, l), we apply the following update if $H(\theta; x, l)$ (and the sub-gradient) is non-zero:

$$\theta' \leftarrow \theta + \eta\left(-\lambda\theta + \phi(l, x) - \alpha\phi(l^\star, x) - (1-\alpha)\frac{1}{M}\sum_i \frac{1}{|\mathcal{L}_i|}\sum_{l_i \in \mathcal{L}_i} \phi(l^\star(l_i), x)\right).$$

Above, η is a learning rate parameter, $l^\star = \arg\max_{l'} \theta_x(l')$ is the highest scoring assignment and $l^\star(l_i) = \arg\max_{l':l'_i=l_i} \theta_x(l')$ are highest scoring assignments constrained to l_i for part i. The key distinguishing feature of this update as compared to structured perceptron is that it subtracts features included in all max-marginal assignments $l^\star(l_i)$[1].

The stages of the cascade are learned sequentially, from coarse to fine, and each has a different θ and $\mathcal{L}_i$ for each part, as well as α. The states of the next level are simply refined versions of the states that have not been pruned. We describe the refinement structure of the cascade in Section 5. In the end of a coarse-to-fine cascade we are left with a small, sparse set of states that typically contains the groundtruth states or states relatively close to them—in practice we are left with around 500 states per part, and 95% of the time we retain a state the is close enough to be considered a match (see Table 2). At this point we have the freedom to add a variety of complex unary and pairwise part interaction features involving geometry, appearance, and compatibility with perceptual grouping principles which we describe in Section 4.

Why not just detector-based pruning? A naive approach used in a variety of applications is to simply subsample states by thresholding outputs of part or sparse feature detectors, possibly combined with non-max suppression. Our approach, based on pruning on max-marginal values in a first-order model, is more sophisticated: for articulated parts-based models, strong evidence from other parts can keep a part which has weak individual evidence, and would be pruned using only detection scores. The failure of prefiltering part locations in human pose estimation is also noted by [6], and serves as the primary justification

[1] Note that because (5) is λ-strongly convex, if we chose $\eta_t = 1/(\lambda t)$ and add a projection step to keep θ in a closed set, the update would correspond to the Pegasos update with convergence guarantees of $\tilde{O}(1/\epsilon)$ iterations for ϵ-accurate solutions [18]. In our experiments, we found the projection step made no difference and used only 2 passes over the data, with η fixed.

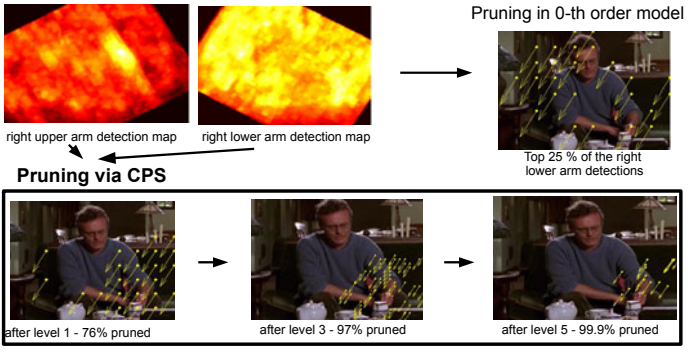


Fig. 3. Upper right: Detector-based pruning by thresholding (for the lower right arm) yields many hypotheses far way from the true one. Lower row: The CPS, however, exploits global information to perform better pruning.

for their use of the dense classical PS. This is illustrated in Figure 3 on an example image from [5].

4 Features

The introduced CPS model allows us to capture appearance, geometry and shape information of parts and pairs of parts in the final level of the cascade—much richer than the standard geometric deformation costs and texture filters of previous PS models [2,4,5,6]. Each part is modeled as a rectangle anchored at the part joint with the major axis defined as the line segment between the joints (see Figure 2). For training and evaluation, our datasets have been annotated only with this part axis.

Shape: We express the shape of limbs via region and contour information. We use contour cues to capture the notion that limbs have a long smooth outline connecting and supporting both the upper and lower parts. Region information is used to express coarse global shape properties of each limb, attempting to express the fact the limbs are often supported by a roughly rectangular collection of regions—the same notion that drives the bottom-up hypothesis generation in [10,17].

Shape/Contour: We detect long smooth contours from sequences of image segmentation boundaries obtained via NCut [24]. We define a graph whose nodes are all boundaries between segments with edges linking touching boundaries. Each contour is a path in this graph (see Fig. 4, middle left). To reduce the number of possible paths, we restrict ourselves to all shortest paths. To quantify the smoothness of a contour, we compute an angle between each two touching segment boundaries[2]. The smoothness of a contour is quantified as the maximum

[2] This angle is computed as the angle between the lines fitted to the segment boundary ends, defined as one third of the boundary.

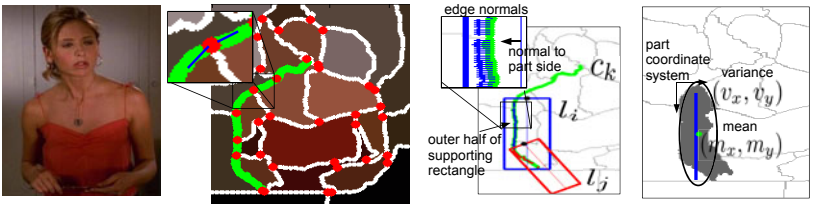


Fig. 4. Left: input image; Middle left: segmentation with segment boundaries and their touching points in red. Middle right: contour edges which support part l_i and have normals which do not deviate from the part axis normal by more than ω. Right: first and second order moments of the region lying under the major part axis.

angle between boundaries along this contour. Finally, we find among all shortest paths those whose length exceeds ℓ_{th} pixels and whose smoothness is less then s_{th} and denote them by $\{c_1, \dots c_m\}$.[3]

We can use the above contours to define features for each pair of lower and upper arms, which encode the notion that those two parts should share a long smooth contour, which is parallel and close to the part boundaries. For each arm part l_i and a contour c_k we can estimate the edges of c_k which lie inside one of the halves of the supporting rectangle of l_i and whose edge normals build an angle smaller than ω with the normal of the part axis (see Fig. 4, right). We denote the number of those edges by $q_{ik}(\omega)$. Intuitively, a contour supports a limb if it is mostly parallel and enclosed in one of the limb sides, i.e. the value $q_{ik}(\omega)$ is large for small angles ω. A pair of arm limbs l_i, l_j should have a high score if both parts are supported by a contour c_k, which can be expressed as the following two scores

$$\text{cc}^{(1)}_{ijk}(\omega, \omega') = \frac{1}{2}\left(\frac{q_{ik}(\omega)}{h_i} + \frac{q_{jk}(\omega')}{h_j}\right) \quad \text{and} \quad \text{cc}^{(2)}_{ijk}(\omega, \omega') = \min\left\{\frac{q_{ik}(\omega)}{h_i}, \frac{q_{jk}(\omega')}{h_j}\right\}$$

where we normalize q_{ik} by the length of the limb h_i to ensure that the score is in $[0, 1]$. The first score measures the overall support of the parts, while the second measures the minimum support. Hence, for l_i, l_j we can find the highest score among all contours, which expresses the highest degree of support which this pair of arms can receive from any of the image contours:

$$\text{cc}^{(t)}_{ij}(\omega, \omega') = \max_{k \in \{1,\dots,m\}} \text{cc}^{(t)}_{ijk}(\omega, \omega'), \quad \text{for} \quad t \in \{1, 2\}$$

By varying the angles ω and ω' in a set of admissible angles Ω defining parallelism between the part and the contour, we obtain $|\Omega|^2$ contour features[4].

Shape/Region Moments: We compute the first and second order moments of the segments lying under the major part axis (see Fig. 4, right)[5] to coarsely

[3] We set $\ell_{\text{th}} = 60$ pixels, $s_{\text{th}} = 45°$ resulting in 15 to 30 contours per image.
[4] We set $\Omega = \{10°, 20°, 30°\}$, which results in 18 features for both scores.
[5] We select segments which cover at least 25% of the part axis.

express shape of limb hypotheses as a collection of segments, R_{l_i}. To achieve rotation and translation invariance, we compute the moments in the part coordinate system. We include convexity information $|conv(R_{l_i})|/|R_{l_i}|$, where $conv(\cdot)$ is the convex hull of a set of points, and $|R_{l_i}|$ is the number of points in the collection of segments. We also include the number of points on the convex hull, and the number of part axis points that pass through R_{l_i} to express continuity along the part axis.

Appearance/Texture: Following the edge-based representation used in [19], we model the appearance the body parts using Histogram of Gradient (HoG) descriptor. For each of the 6 body parts – head, torso, upper and lower arms – we learn an individual Gentleboost classifier [20] on the HoG features using the Limbs Annotated in Movies Dataset[6].

Appearance/Color: As opposed to HoG, color drastically varies between people. We use the same assumptions as [21] and build color models assuming a fixed location for the head and torso at run-time for each image. We train Adaboost classifiers using these pre-defined regions of positive and negative example pixels, represented as RGB, Lab, and HSV components. For a particular image, a 5-round Adaboost ensemble [22] is learned for each color model (head, torso) and reapplied to all the pixels in the image. A similar technique is also used by [23] to incorporate color. Features are computed as the mean score of each discrimintative color model on the pixels lying in the rectangle of the part.

We use similarity of appearance between lower and upper arms as features for the pairwise potentials of CPS. Precisely, we use the χ^2 distance between the color histograms of the pixels lying in the part support. The histograms are computed using minimum-variance quantization of the RGB color values of each image into 8 colors.

Geometry: The body part configuration is encoded in two set of features. The location (l_{ix}, l_{iy}) and orientation (l_{iu}, l_{iv}), included in the state of a part, are used added as absolute location prior features. We express the relative difference between part l_i its parent l_j in the coordinate frame of the parent part as $T_{ij}(l_i) - l_j$. Note we could introduce second-order terms to model a quadratic deformation cost akin to the classical PS, but we instead adopt more flexible binning or boosting of these features (see Section 5).

5 Implementation Details

Coarse-to-Fine Cascade. While our fine-level state space has size $80 \times 80 \times 24$, our first level cascade coarsens the state-space down to $10 \times 10 \times 12 = 1200$ states per part, which allows us to do exhaustive inference efficiently. We always train and prune with $\alpha = 0$, effectively throwing away half of the states at each stage. After pruning we double one of the dimensions (first angle, then the minimum of width or height) and continue (see Table 2). In the coarse-to-fine stages we only use standard PS features. HoG part detectors are run once over the original

[6] LAMDa is available at `http://vision.grasp.upenn.edu/video`

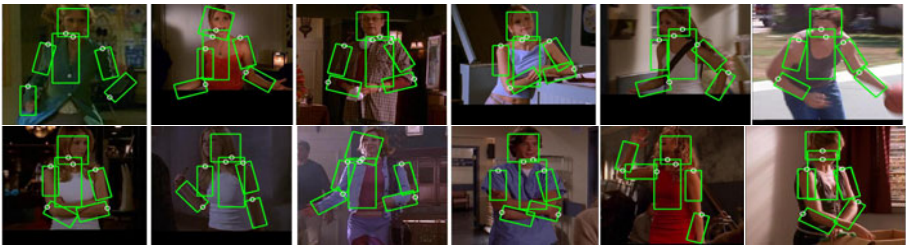


Fig. 5. Examples of correctly localized limbs under different conditions (low contrast, clutter) and poses (different positions of the arms, partial self occlusions)

state space, and their outputs are resized to for features in coarser state spaces. We also use the standard relative geometric cues as described in Sec. 4. We bin the values of each feature uniformly, which adds flexibility to the standard PS model—rather than learning a mean and covariance, multi-modal pairwise costs can be learned.

Sparse States, Rich Features. To obtain segments, we use NCut [24]. For the contour features we use 30 segments and for region moments – 125 segments. As can be seen in Table 2, the coarse-to-fine cascade leaves us with roughly 500 hypotheses per part. For these hypotheses, we generate all features mentioned in Sec. 4. For pairs of part hypotheses which are farther than 20% of the image dimensions from the mean connection location, features are not evaluated and an additional feature expressing this condition is added to the feature set. We concatenate all unary and pairwise features for part-pairs into a feature vector and learn boosting ensembles which give us our pairwise clique potentials[7]. This method of learning clique potentials has several advantages over stochastic subgradient learning: it is faster to train, can determine better thresholds on features than uniform binning, and can combine different features in a tree to learn complex, non-linear interactions.

6 Experiments

We evaluate our approach on the publicly available Buffy The Vampire Slayer v2.1 and PASCAL Stickmen datasets [21]. We use the upper body detection windows provided with the dataset as input to localize and scale normalize the images before running our experiments as in [21,5,6]. We use the usual 235 Buffy test images for testing as well as the 360 detected people from PASCAL stickmen. We use the remaining 513 images from Buffy for training and validation.

Evaluation Measures. The typical measure of performance on this dataset is a matching criteria based on both endpoints of each part (e.g., matching the elbow and the wrist correctly): A limb guess is correct if the limb endpoints are

[7] We use OpenCV's implementation of Gentleboost and boost on trees of depth 3, setting the optimal number of rounds via a hold-out set.

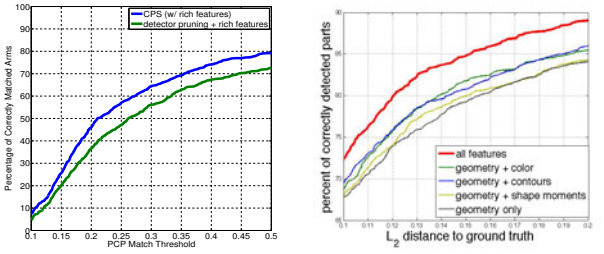


Fig. 6. Left: PCP curves of our cascade method versus a detection pruning approach, evaluated using PCP on arm parts (see text). **Right:** Analysis of incorporating individual types of features into the last stage of our system.

on average within r of the corresponding groundtruth segments, where r is a fraction of the groundtruth part length. By varying r, a performance curve is produced where the performance is measured in the percentage of correct parts (PCP) matched with respect to r.

Overall system performance. As shown in Table 1, we perform comparably with the state-of-the-art on all parts, improving over [25] on upper arms on both datasets and significantly outperforming earlier work. We also compare to a much simpler approach, inspired by [16] (detector pruning + rich features): We prune by thresholding each unary detection map individually to obtain the same number of states as in our final cascade level, and then apply our final model with rich features on these states. As can be seen in Figure 6/left, this baseline performs significantly worse than our method (performing about as well as a standard PS model as reported in [25]). This makes a strong case for using max-marginals (e.g., a global image-dependent quantity) for pruning, as well as learning how to prune safely and efficiently, rather than using static thresholds on individual part scores as in [16].

Our previous method [25] is the only other PS method which incorporates image information into the pairwise term of the model. However it is still an exhaustive inference method. Assuming all features have been pre-computed, inference in [25] takes an average of 3.2 seconds, whereas inference using the sparse set of states in the final stage of the cascade takes on average 0.285 seconds—a speedup of 11.2x[8].

In Figure 6/right we analyze which features are most effective, measured in L_2 distance to the groundtruth state, normalized by the groundtruth length of the part. We start only with the basic geometry and unary HoG detector features available to basic PS systems, and add different classes of features individually. Skin/torso color estimation gives a strong boost in performance, which is consistent with the large performance boost that the results in [21] obtained over their previous results [12]. Using contours instead of color is nearly as effective. The

[8] Run on an Intel Xeon E5450 3.00GHz CPU with an $80 \times 80 \times 24$ state space averaged over 20 trials. [25] uses MATLAB's optimized fft function for message passing.

Table 1. Comparison to other methods at $PCP_{0.5}$. See text for details. We perform comparably to state-of-the-art on all parts, improving on upper arms.

method	torso	head	upper arms	lower arms	total
Buffy					
Andriluka et al. [6]	90.7	95.5	79.3	41.2	73.5
Eichner et al. [21]	98.7	97.9	82.8	59.8	80.1
APS [25]	100	100	91.1	65.7	85.9
CPS (ours)	100	96.2	95.3	63.0	85.5
Detector pruning	99.6	87.3	90.0	55.3	79.6
PASCAL stickmen					
Eichner et al. [21]	97.22	88.60	73.75	41.53	69.31
APS [25]	100	98.0	83.9	54.0	79.0
CPS (ours)	100	90.0	87.1	49.4	77.2

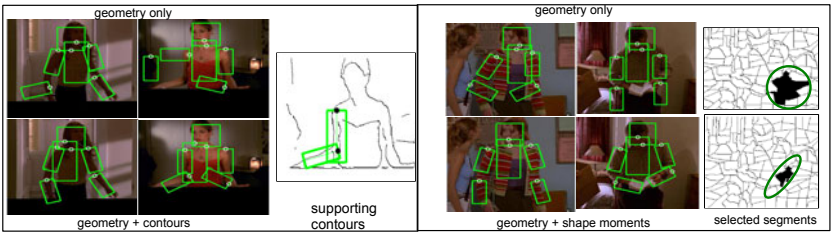


Fig. 7. Detections with geometry (top) and with additional cues (bottom). Left: contour features support arms along strong contours and avoid false positives along weak edges. Right: after overlaying the part hypothesis on the segmentation, the incorrect one does not select an elongated set of segments.

features combine to outperform any individual feature. Examples where different cues help are shown in Figure 7.

Coarse-to-fine Cascade Evaluation: In Table 2, we evaluate the drop in performance of our system after each successive stage of pruning. We report PCP scores of the best possible as-yet unpruned state left in the original space. We choose a tight $PCP_{0.2}$ threshold to get an accurate understanding whether we have lost well-localized limbs. As seen in Table 2, the drop in $PCP_{0.2}$ is small and linear, whereas the pruning of the state space is exponential—half of the states are pruned in the first stage. As a baseline, we evaluate the simple detector-based pruning described above. This leads to a significant loss of correct hypotheses, to which we attribute the poor end-system performance of this baseline (in Figure 6 and Table 1), even after adding richer features.

Future work: The addition of more powerful shape-based features could further improve performance. Additional levels of pruning could allow for (1) faster inference, (2) inferring with higher-order cliques to, e.g., express compatability between left and right arms or (3) incorporating additional variables into the state space—relative scale of parts to model foreshortening, or occlusion variables. Finally, our approach can be naturally extended to pose estimation in video where the cascaded models can be coarsened over space and time.

Table 2. For each level of the cascade we present the reduction of the size of the state space after pruning each stage and the quality of the retained hypotheses measured using $PCP_{0.2}$. As a baseline, we compare to pruning the same number of states in the HoG detection map (see text).

cascade stage	state dimensions	# states in the original space	# states in the pruned space	state space reduction %	$PCP_{0.2}$ arms oracle
0	10x10x12	153600	1200	00.00	—
1	10x10x24	72968	1140	52.50	54
3	20x20x24	6704	642	95.64	51
5	40x40x24	2682	671	98.25	50
7	80x80x24	492	492	99.67	50
detection pruning	80x80x24	492	492	99.67	44

References

1. Fischler, M., Elschlager, R.: The representation and matching of pictorial structures. IEEE Transactions on Computers 100, 67–92 (1973)
2. Felzenszwalb, P., Huttenlocher, D.: Pictorial structures for object recognition. IJCV 61, 55–79 (2005)
3. Fergus, R., Perona, P., Zisserman, A.: A sparse object category model for efficient learning and exhaustive recognition. In: Proc. CVPR (2005)
4. Ramanan, D., Sminchisescu, C.: Training deformable models for localization. In: CVPR, pp. 206–213 (2006)
5. Ferrari, V., Marin-Jimenez, M., Zisserman, A.: Progressive search space reduction for human pose estimation. In: Proc. CVPR (2008)
6. Andriluka, M., Roth, S., Schiele, B.: Pictorial structures revisited: People detection and articulated pose estimation. In: Proc. CVPR (2009)
7. Fleuret, G., Geman, D.: Coarse-to-Fine Face Detection. IJCV 41, 85–107 (2001)
8. Viola, P., Jones, M.: Robust real-time object detection. IJCV 57, 137–154 (2002)
9. Lan, X., Huttenlocher, D.: Beyond trees: Common-factor models for 2d human pose recovery. In: Proc. ICCV, pp. 470–477 (2005)
10. Mori, G., Ren, X., Efros, A., Malik, J.: Recovering human body configurations: Combining segmentation and recognition. In: CVPR (2004)
11. Ramanan, D.: Learning to parse images of articulated bodies. In: NIPS (2006)
12. Ferrari, V., Marin-Jimenez, M., Zisserman, A.: Pose search: retrieving people using their pose. In: Proc. CVPR (2009)
13. Weiss, D., Taskar, B.: Structured prediction cascades. In: Proc. AISTATS (2010)
14. Carreras, X., Collins, M., Koo, T.: TAG, dynamic programming, and the perceptron for efficient, feature-rich parsing. In: Proc. CoNLL (2008)
15. Petrov, S.: Coarse-to-Fine Natural Language Processing. PhD thesis, University of California at Bekeley (2009)
16. Felzenszwalb, P., Girshick, R., McAllester, D.: Cascade Object Detection with Deformable Part Models. In: Proc. CVPR (2010)
17. Srinivasan, P., Shi, J.: Bottom-up recognition and parsing of the human body. In: ICCV 2005, pp. 824–831. IEEE Computer Society, Los Alamitos (2007)
18. Shalev-Shwartz, S., Singer, Y., Srebro, N.: Pegasos: Primal estimated sub-gradient SOlver for SVM. In: Proc. ICML (2007)

19. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object Detection with Discriminatively Trained Part Based Models. In: PAMI (2008)
20. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: A statistical view of boosting. The Annals of Statistics 28, 337–374 (2000)
21. Eichner, M., Ferrari, V.: Better appearance models for pictorial structures. In: Proc. BMVC (2009)
22. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. JCSS 55, 119–139 (1997)
23. Ramanan, D., Forsyth, D., Zisserman, A.: Strike a pose: Tracking people by finding stylized poses. In: Proc. CVPR, vol. 1, p. 271 (2005)
24. Cour, T., Benezit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. In: Proc. CVPR (2005)
25. Sapp, B., Jordan, C., Taskar, B.: Adaptive pose priors for pictorial structures. In: CVPR (2010)

State Estimation in a Document Image and Its Application in Text Block Identification and Text Line Extraction

Hyung Il Koo and Nam Ik Cho

INMC, Dept. of EECS, Seoul National University
hikoo@ispl.snu.ac.kr, nicho@snu.ac.kr

Abstract. This paper proposes a new approach to the estimation of document states such as interline spacing and text line orientation, which facilitates a number of tasks in document image processing. The proposed method can be applied to spatially varying states as well as invariant ones, so that general cases including images of complex layout, camera-captured images, and handwritten ones can also be handled. Specifically, we find CCs (Connected Components) in a document image and assign a state to each of them. Then the states of CCs are estimated using an energy minimization framework, where the cost function is designed based on frequency domain analysis and minimized via graph-cuts. Using the estimated states, we also develop a new algorithm that performs text block identification and text line extraction. Roughly speaking, we can segment an image into text blocks by cutting the distant connections among the CCs (compared to the estimated interline spacing), and we can group the CCs into text lines using a bottom-up grouping along the estimated text line orientation. Experimental results on a variety of document images show that our method is efficient and provides promising results in several document image processing tasks.

Keywords: document image processing, state estimation, graph cuts, text block identification, text line extraction.

1 Introduction

Text block identification and text line extraction are fundamentally important steps for OCR (Optical Character Recognition), and they are also essential for the rectification of camera-captured document images [1,2,3,4,5,6]. However, most research in this area has assumed scanned documents [1,7,8,9,10] and the applications to camera-captured images were limited to relatively simple layout and text-abundant cases [2,4,11]. In order to widen the area of valuable document processing tools (such as OCR and TTS for visually impaired, automatic translation of books and street signs, etc) to the camera-captured inputs, we propose a novel document state estimation algorithm and present its application in text block identification and text line extraction, where the state means line spacing, orientation, and other parameters describing the local properties of

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 421–434, 2010.

text region. Examples of input and output of our algorithm are shown in Fig. 1-(a) and (f). As can be seen, camera-captured images suffer from perspective distortion, geometric distortion, uneven illumination, motion blur, un-focussed blur, non-textual objects, and possibly cluttered background.

1.1 Our Method

Our method consists of two parts. In the former part, we estimate interline spacing and text line orientation for each Connected Component (CC), where we call two properties as the state of a CC. This step may correspond to a scale selection step in feature detection methods [12]. As the scale selection is important in detecting features from unknown measurement data, the state estimation is essential for unconstrained document image processing. For example, a simple problem to determine whether two adjacent CCs are in a same word or not may be ambiguous unless we know their states. Nevertheless, there is little research on this problem in camera based methods. It is probably because appropriate states for analysis may be known a priori in controlled situations [3,4,5]. However, we believe that the state estimation is an essential step for camera-captured image processing not only for the theoretical aspects but also for a practical system that can be demonstrated in uncontrolled environments (unknown character size, page curl, shot angle, and distance). In the latter part of our method, we develop a method that identifies text blocks and extracts text lines using the estimated states. Especially, the text line extraction method is based on a bottom-up grouping as commonly used in other related works [1,3,5,13]. However, unlike the other works, our method is largely free from conventional drawbacks due to the estimated states.

State estimation of CCs. The idea of assigning states can be found in the literature [1,9,10,15]. In docstrum [1], nearest neighbor (NN) angle histogram and NN distance histogram are computed from the geometric relationship between K-nearest units. From the histograms, they estimated the orientation, interline spacing, and within-line spacing. Then, a bottom-up approach is adopted to cluster CCs into words, text lines, and blocks. Due to the state estimation, the algorithm can effectively accomplish skew estimation and page segmentation [7], however, the method cannot handle the spatially varying cases [13]. It is because the method assumes fixed states (i.e., not spatially-varying) and the same rules using the same parameters are applied to the whole image. Related works can be found in [7,9,10,15]. Since camera-captured images, handwritten ones, and documents having complex layout have spatially varying properties, we assume that each CC has its own state. Also, we formulate the state estimation problem as an energy minimization problem. In designing the energy function, we consider a neighborhood system induced by Delaunay triangulation [14] and a data term is designed to exploit the periodical property of text lines.

Text block identification and text line detection. For text block identification and text line extraction, we first segment a graph formed by Delaunay

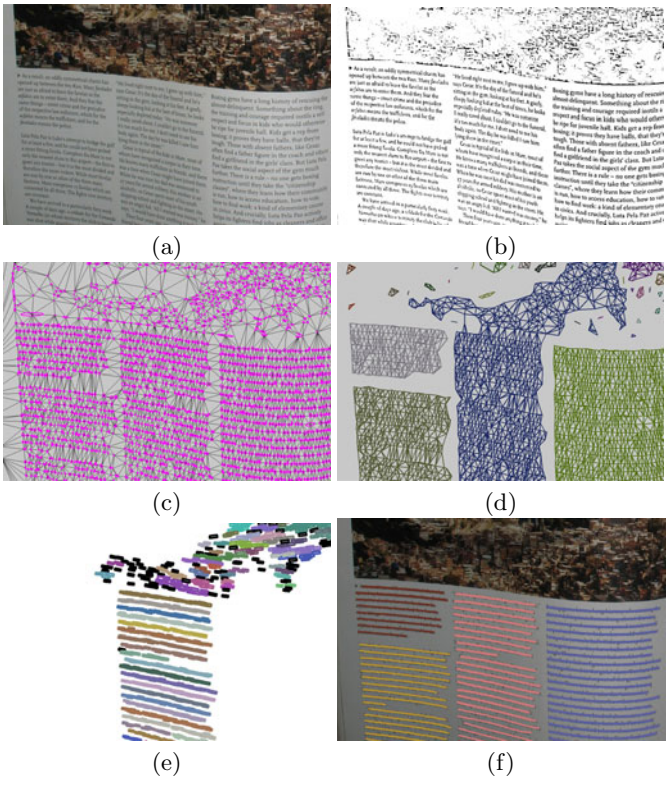


Fig. 1. Illustration of our algorithm, (a) Input of our algorithm, (b) Binarization result, (c) Super-pixel representation and Delaunay triangulation [14], (d) Detected text blocks. See Section 3.1 for details, (e) Our bottom-up grouping result. See Section 3.2 for details, (f) Our final result.

triangulation of CCs (Fig. 1-(c)) into subgraphs (Fig. 1-(d)) by removing long edges that connect the CCs. Then, we cluster the CCs into text lines using a bottom-up approach. It is noted that the conventional bottom-up approaches are sensitive to input variations such as language, character size, and page curl [16,11] since they required heuristic rules, artificial parameters, and training process [5,13,3]. However, our method can be robust to the variations by using the estimated scale and orientation. Compared to recently developed text line extraction methods that adopt general image segmentation techniques [17,11], our method is more efficient and detects text lines in a scale/orientation invariant manner.

However, like other methods, our method also suffers from non-textual objects as shown in Fig. 1-(e). For non-textual object rejection, a training based method was proposed in [8] for classifying each block into printed text, handwriting, or noise. Although their results are very convincing, it is not clear how to construct a training set that achieves robustness to language variation, poor image quality,

and complex layout. Since the noise that smears text region [7,8] is seldom observed in camera-captured images of printed material, we assume that non-textual objects take place distant from text blocks. Then we can reject non-textual objects using the properties of clusters. Precisely, we assume that (1) a cluster in text region tends to be curvilinear, and (2) a cluster in non-textual region is isolated (represented as black rectangles) or it may be non-curvilinear as can be seen in Fig. 1-(e). Using these properties, we formulate non-textual object rejection as a labeling problem with an energy minimization approach. After the inference, we remove non-textual objects, and refine text blocks and text lines. The result is illustrated in Fig. 1-(f).

2 The State Estimation of CCs

In this section, we explain our state estimation method based on an energy minimization framework. This section consists of binarization, CC construction, energy formulation, and its minimization that gives the state of each CC.

2.1 Binarization and CC Construction

The first step of our algorithm is the binarization of a gray image I. Our binarization method is based on the retinex filtering which is efficient and robust to uneven illumination:

$$B_s = \begin{cases} 1 & I_s < \mu_1 \times G_s \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where I_s is the intensity at pixel s, $G(\sigma) * I$ is a Gaussian filtered image of I, and $G_s = (G(\sigma) * I)_s$ [18]. However, since it produces a number of spurious responses on dark and homogeneous region, we introduce an additional condition that suppresses responses on homogeneous region:

$$|I_s - G_s| > \mu_2 \tag{2}$$

for $B_s = 1$. From the binary image $\{B_s\}$, we extract CCs of '1' using an 8-neighborhood system. In this process, we suppress small CCs (containing less than 10 pixels) and large CCs (containing more than 3000 pixels) for the removal of noisy ones. We denote the set of extracted CCs as $\mathcal{P}$.

2.2 Energy Formulation

We assign a state to every site $p \in \mathcal{P}$, and denote the state as $f_p = (s_p, \theta_p)$, where s_p is the interline spacing between neighboring text lines and θ_p is the orientation of a text line where p belongs. The estimation problem is formulated as an energy minimization problem whose energy function is given by

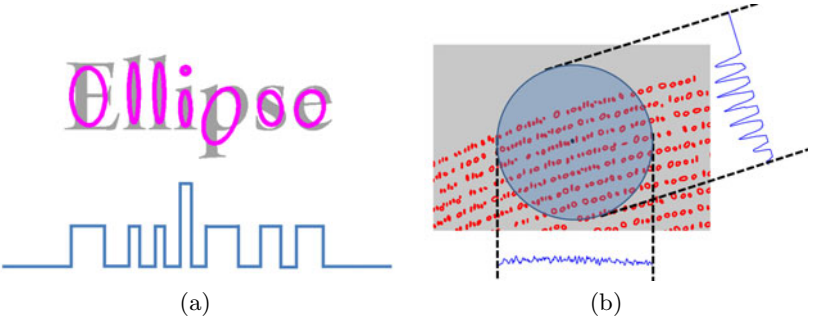
$$E(\{f_p\}) = \sum_{p \in \mathcal{P}} V_p(f_p) + \sum_{(p,q) \in \mathcal{E}} V_{p,q}(f_p, f_q) \tag{3}$$

where $V_p(f_p)$ is a data term reflecting local observation, $V_{p,q}(f_p, f_q)$ is a pairwise potential reflecting label smoothness, and $\mathcal{E}$ is a set of edges.

2.3 The Design of $V_p(f_p)$

For the design of $V_p(f_p)$, we first explain our projection method, and frequency domain analysis will be followed.

Super-pixel approximation. Since pixel based approaches are computationally demanding in analyzing local patterns, we reduce complexity by using super-pixels. Precisely, we compute the mean vector (x_p, y_p) and the covariance matrix Σ_p of pixels in the p-th CC. The covariance matrix is decomposed into $\Sigma_p = \sigma_1 v_1 v_1^T + \sigma_2 v_2 v_2^T$ where $\sigma_1 > \sigma_2$ are eigenvalues, v_1 and v_2 are eigenvectors. Using the decomposition, the CC is approximated to an ellipse (whose minor and major axes are v_1 and v_2 respectively) as illustrated in Fig. 2-(a). In this process, ellipses showing large eccentricity ($\frac{\sigma_1}{\sigma_2} > 15$) are also removed. Then we define a projected signal, which is the number of ellipse on the line of projection as illustrated in Fig. 2-(a). Fig. 2-(b) shows an example of projecting the super-pixels in a circle into some directions.



(a) (b)

Fig. 2. (a) Ellipse approximation of CCs and its projection, (b) Ellipse approximation of CCs and its projected signals into two directions

Data term based on frequency domain analysis. As illustrated in Fig. 2-(b), when CCs around a site p are projected to the normal direction to a text line, a periodic pattern (whose period is the interline spacing s_p) is observed. From the observation, we design $V_p(f_p)$ so that it decreases as the periodicity of projected signal is increasing. For this, we first obtain a projected signal $x(n)$ by projecting CCs to the orientation of θ_p, and its DFT $X_N(k)$ is computed: $X_N(k) = \sum_{n=0}^{N-1} x(n) \exp\left(-j\frac{2\pi kn}{N}\right)$. The normalized energy of a signal of period $\frac{N}{k}$ is given by

$$\frac{|X_N(k)|^2 + |X_N(2k)|^2 + \cdots}{|X_N(0)|^2 + |X_N(1)|^2 + |X_N(2)|^2 + \cdots} \simeq \frac{|X_N(k)|^2}{|X_N(0)|^2} \tag{4}$$

where the numerator of left hand side is the energy of repeating component ($T = \frac{N}{k}$) and the denominator is the overall energy of $x(n)$ [19]. Moreover, we verified through experiment that this can be replaced as the magnitude of first

harmonic over the DC term as shown in the right-hand side of (4). Based on this measure of periodicity, $V_p(f_p)$ is defined as

$$V_p(f_p) = -\log \frac{|X_N(k)|^2}{|X_N(0)|^2}. \tag{5}$$

Finally, we have to choose (s_p, N_p, k_p) satisfying $\frac{N_p}{k_p} = s_p$. There are several factors to be considered. For the good localization in frequency domain, a large N is desirable. On the other hand, a large N is not good at handling spatially varying states. We also have to consider computational complexity. Considering these factors, we select 10 scales from $12.8 \leq s_p \leq 128$ and they are summarized in Table 1. We also quantize orientations into $D = 32$ steps:

$$\theta_p \in \left\{ i \times \frac{\pi}{D} \middle| i = 0, 1, \ldots, D-1 \right\}. \tag{6}$$

In summary, $V_p(f_p)$ for label $f_p = (s_p, \theta_p)$ is computed as follows.

- CCs around the p-th CC are projected to the line whose orientation is θ_p, resulting $x(n)$. In the projection, the size of window is determined according to the Table 1.

Table 1. Discrete levels of interline spacing (s_p) used in our algorithm

Discrete levels (s_p)	12.8	16.0	21.3	25.6	32.0	42.7	51.2	64.0	85.3	128.0
N_p	64	64	64	128	128	128	256	256	256	256
k_p	5	4	3	5	4	3	5	4	3	2

- When there are only small number of CCs (i.e., $x(n) \leq 3$ for all n) or $|X_{N_p}(k_p)|$ is not a local maximum, we set $V_p(f_p) = \epsilon$.
- Otherwise, the data cost is given by

$$V_p(f_p) = -\log \frac{|X_{N_p}(k_p)|^2}{|X_{N_p}(0)|^2}. \tag{7}$$

2.4 Pairwise Potential

For a neighborhood system, we adopt Delaunay triangulation [14] and our pairwise potential is given by

$$V_{p,q}(f_p, f_q) = \mu(f_p, f_q) \times \exp\left(-\frac{k \times d_{pq}^2}{(s_p^2 + s_q^2)} \right) \tag{8}$$

where d_{pq} is the Euclidean distance between the site p and q. Since $d_{pq}/\sqrt{s_p^2 + s_q^2}$ can be considered as an intrinsic distance (i.e., invariant to camera settings,

distance between documents and camera, and so on) between two CCs, the cost function allows label discontinuities between distant sites while imposing smoothness constraints on nearby ones. Moreover, in order to allow small amount of label discontinuities (which is common in camera-captured documents), $\mu(f_p, f_q)$ is defined as

$$\mu(f_p, f_q) = \begin{cases} 0 & f_p = f_q \\ \lambda_1 & |f_p - f_q| \leq 3 \\ \lambda_2 & \text{otherwise} \end{cases} \tag{9}$$

where $|f_p - f_q|$ is the label distance defined as the sum of orientation difference and scale difference ($\lambda_1 < \lambda_2$).

2.5 Optimization

In (3), the number of sites ($|\mathcal{P}|$) is usually up to $20,000$ and the number of labels is 32×10. We optimize the cost function using *Expansion move* algorithm [20].

3 Text Block Identification and Text Line Extraction

In this section, we explain the latter part of our algorithm. From the estimate states, (1) we segment a document image into blocks by removing long edges, (2) each block is decomposed into clusters, and (3) we reject non-textual clusters by considering the curvilinearity of each cluster and neighboring relations. Finally, (4) we refine text blocks and text lines.

3.1 Page Segmentation

For page segmentation, we remove perpendicular edges (which are perpendicular to text lines) satisfying $d_{pq} \geq \epsilon_1 \times \min(s_p, s_q)$, and we remove parallel edges satisfying $d_{pq} \geq \epsilon_2 \times \min(s_p, s_q)$. Since the edges connecting two vertically adjacent regions are usually longer than edges connecting horizontally adjacent regions, we can achieve more accurate segmentation by considering orientation as well as interline spacing. Two constants are determined according to conventional layout: $(\epsilon_1, \epsilon_2) = (1.2, 0.9)$.

However, this method may suffer from perspective contraction, coarse quantization of interline spacing, and noise. That is, a CC on text region and another CC on a picture region may be linked as Fig. 1-(d). Therefore, non-textual object rejection should be applied. Since non-textual object rejection is closely related with our bottom-up grouping method, we explain the method in the next section and the explanation on non-textual object rejection will be followed.

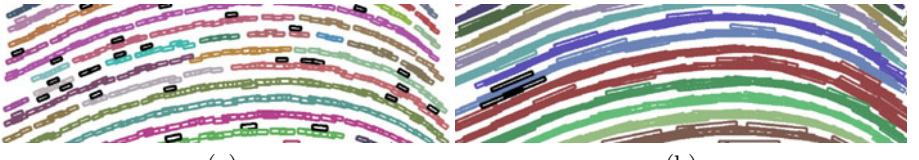
Skew correction. After page segmentation, we first find the dominant angle of a text block by using a voting method and compensate the skew in order to represent a text line as a form of $y = f(x)$ without numerical instability.

3.2 Bottom-Up Grouping

For grouping, we draw a rectangle for each CC, whose size is $ws_p \times hs_p$, its center (x_p, y_p), and rotated by θ_p(+ text block skew). Then, each connected region corresponds to a word or a text line as can be seen in Fig. 3. However, a single choice of (w, h) is not adequate. When small (w, h) is used, a text line may be partitioned into several clusters (over-segmentation of a text line) as Fig. 3-(a). On the other hand, more than one line may be merged into a single cluster (under-segmentation of a text line) when large (w, h) is used as shown in Fig. 3-(b). Therefore, we develop a method that incrementally increases w value (fixing $h = 0.25$). First, we group CCs into clusters using $w_1 = 0.8$, resulting a set of clusters $\mathcal{W}$. Then, two clusters C_i, $C_j \in \mathcal{W}$ are merged into a new cluster (i.e., C_i and C_j in $\mathcal{W}$ are replaced with $C_i \cup C_j$) when three conditions are satisfied:

1. Two clusters are connected when a new w_i is used.
2. The overlap of two supports (x-domain) is less than 10% of their length.
3. A new cluster $(C_i \cup C_j)$ is still a curvilinear one (the detailed explanation of this condition will be followed in the next section).

Intuitively, the second and third conditions prevent the merging of neighboring text lines. We use $w_2 = 1.0$ and $w_3 = 1.2$. Fig 1-(e) shows our bottom-up grouping result.



(a) (b)

Fig. 3. (a) When we use small (w, h), a text line can be segmented into several small clusters (over-segmentation), (b) If we use large (w, h), more than one line can be merged into a single cluster (under-segmentation)

3.3 Curvilinearity Measure

For the curvilinearity measure of a cluster $C \in \mathcal{W}$, we define the fitting error of C and the scale of C. The fitting error of C is defined as

$$\eta(C) = \sqrt{\frac{1}{|C|} \times \min_f \sum_{p \in C} \left| y'_p - f(x'_p) \right|^2} \tag{10}$$

where $|C|$ is the number of CCs in C, the degree of polynomial f is determined according to $|C|$ (from first to fourth order polynomials), and (x'_p, y'_p) is the rotated point of (x_p, y_p) by the text block skew. Also the scale of C is given by

$$s(C) = \frac{1}{|C|} \sum_{p \in C} s_p. \tag{11}$$

Since $\eta(C)/s(C)$ can be considered as a normalized fitting error, we can measure the curvilinearity by comparing $\eta(C)$ and $s(C)$. For example, $\eta(C) \ll s(C)$ means C is a curvilinear cluster. Experiment results show that most of text lines satisfy $\eta(C) < 0.2 \times s(C)$ and we say that C is curvilinear when it satisfies the inequality.

3.4 Textual/Non-textual Cluster Labeling

Although the proposed curvilinearity test (i.e., $\eta(C) < 0.2 \times s(C)$) provides a good rule to reject non-textual clusters, the performance can be improved by considering neighboring relations. We also formulate the problem as an energy minimization problem. From $\mathcal{W}$, we construct a new graph where each site is an element in $\mathcal{W}$, and denote its label $l_i = 0$ when $C_i \in \mathcal{W}$ is a part of text region, and $l_i = 1$ otherwise. Our energy function is given by

$$E(\{l_i\}) = \sum V_i(l_i) + \lambda_3 \sum e_{ij}\delta(l_i, l_j) \tag{12}$$

where

$$\delta(l_i, l_j) = \begin{cases} 1 & l_i \neq l_j \\ 0 & l_i = l_j. \end{cases} \tag{13}$$

In defining $V_i(l_i)$, we consider two properties that (1) an isolated C_i ($|C_i| \leq 5$) is likely to be a non-textual object and (2) a non-curvilinear C_i is likely to be a non-textual object. Therefore, when $|C_i| \geq 6$, our data term is given by

$$V_i(l_i) = |C_i| \times \begin{cases} \eta(C_i) & l_i = 0 \\ 0.2 \times s(C_i) & l_i = 1. \end{cases} \tag{14}$$

The pairwise term in (12) is derived from (8), and it is given by

$$e_{ij} = \sum_{p \in C_i, q \in C_j, (p,q) \in \mathcal{E}} \exp\left(-\frac{k \times d_{pq}^2}{(s_p^2 + s_q^2)}\right). \tag{15}$$

The cost function is also minimized by graph-cuts [20].

3.5 Text Line Refinement

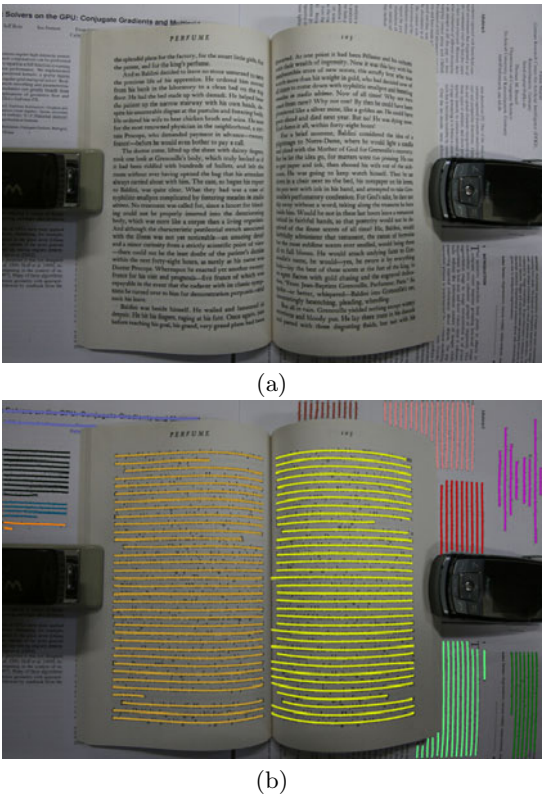
After inference, we remove non-textual clusters. Then, we re-detect text blocks because more than one text block might be merged into a single one via non-textual objects. Also, we re-detect text lines using the procedures presented in Section 3.2. However, at this time, we use a different sequence of $\{w_k\}$ (which will be presented in the experimental section) in order to prevent the over-segmentation of text lines. The final result can be found in Fig 1-(f). As shown in the figure, non-textual objects observed in Fig 1-(e) are successfully rejected.

4 Experimental Results

We have tested our algorithm with more than 300 images including camera-captured ones, and scanned ones. Inputs, binarized results, and experimental results can be found in our website (http://ispl.snu.ac.kr/~hikoo/layout/). Experiments were performed with parameters: $\sigma = 4.5$, $\mu_1 = 0.9$, $\mu_2 = 0.1 \times 255$, $\lambda_1 = 0.4$, $\lambda_2 = 5$, $\lambda_3 = 4$, $k = 0.125$, and $\epsilon = 2.8$. However, we have found that different settings of ϵ_1 and ϵ_2 sometimes provide better results than a default setting ($\epsilon_1 = 1.2$, $\epsilon_2 = 0.9$), and we also present such cases.

4.1 Qualitative Evaluation and Limitations

Fig. 1, Fig. 4, and results in our website show that our algorithm can detect text lines in scale, orientation, and language invariant manner. However, careful observation of them also reveals the limitations of our algorithm. First of all, our method has difficulty in detecting a single-line text because it exploits distribution pattern of text lines. Another limitation is that it is sensitive to motion



(a)

(b)

Fig. 4. Input and output of our algorithm

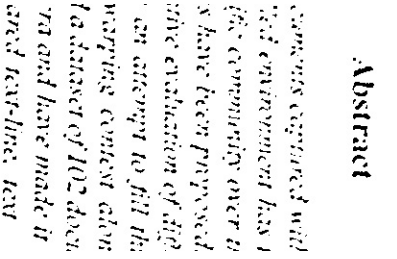
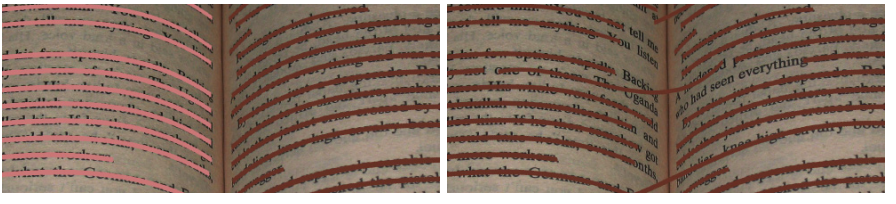


Fig. 5. The binarization result of the right upper part of Fig. 4-(a). As shown in Fig. 4-(b), text lines are successfully extracted even if the binarization performance is not good.



(a) (b)

Fig. 6. *Expansion move* algorithm is sometimes stuck to poor local optima. (a) Our result for $E = 3469$, (b) Our result for $E = 3524$.

blur and shallow field of depth. Blurred inputs result in poor binarized images and they deteriorate the performance of other processes. Although our method tolerates the right upper part of Fig. 4-(a) (whose binarization result is shown in Fig. 5), it fails to handle more blurred inputs as can be found in the left lower part of Fig. 4-(a). The last problem comes from the *Expansion move* algorithm used in the minimization of (3). It sometimes stuck to local minima depending on its initialization. An example for this case is shown in Fig. 6.

4.2 Quantitative Evaluation on Camera Captured Images

For the quantitative evaluation of our method, we have selected 50 images (all of them can be found in our website) in our dataset. Fig. 1-(a) is the cropped version of one of them. For text line refinement, we use $w_1 = 0.8, w_2 = 1.0, w_3 = 1.2, w_4 = 2.0, w_5 = 4.0$, and $w_6 = 6.0$. Experimental results show that 95.7% text blocks among 185 text blocks are correctly detected (we only consider text blocks having more than one text line), and false positive and false negative are less than 2%. In text line detection, our algorithm detects 98.4% text lines correctly (we say that a line is correctly detected when the number of missed characters is less than 4). False positive and false negative are also less than 2%. If we ignore two occluded inputs (See the 25 and 36-th images in our dataset), the results will be improved more than 1%. Our algorithm takes less than 10

seconds in handling 3264×2488 inputs having $10,000$ CCs. Since there is much room for optimization and parallelization (e.g., construction of a data table), we believe that the computational complexity is reasonable.

4.3 Evaluation on Other Dataset

A direct comparison to existing method(s) is not a simple task. It is because our method has been developed to handle complex cases compared to conventional ones [3]. Moreover, our method includes text block identification, which has not been considered in conventional algorithms [3,16]. Therefore, we have applied our method to conventional cases (ICDAR dataset [3]) rather than applying conventional methods to our dataset. In this experiment, we use $(\epsilon_1, \epsilon_2) = (10, 10)$ rather than a default setting. It is because (1) text block segmentation is not an issue in this database (at least in terms of performance evaluation) and (2) our page segmentation method with a default setting is not suitable to detect subtitles or captions as shown in Fig 7-(b). In the text line refinement step, we use $w_1 = 0.8, w_2 = 1.0, w_3 = 1.2$, and $w_4 = 2.0$.

According to the evaluation method in [21,22], the match score is defined as

$$MatchScore(i, j) = \frac{|G_j \cap R_i|}{|G_j \cup R_i|} \tag{16}$$

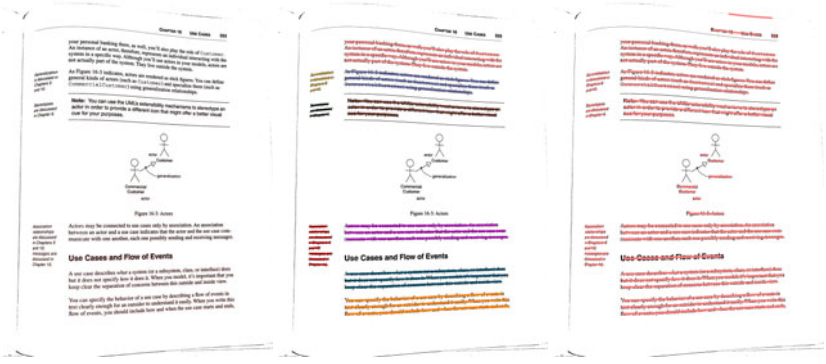
where G_j is the set of all pixels in the j-th ground truth text line and R_i is the set of all pixel in the i-th detected text line. Also, the correct segmentation accuracy [22] is defined as

$$100 \times \frac{\text{the number of matched } (G_j, R_i) \text{ pairs}}{\text{the number of ground truth text lines}} \tag{17}$$

where we consider (G_j, R_i) is a matched pair when $MatchScore(i, j) \geq 0.95$. Experimental results on 102 images show that the correct segmentation accuracy of our method is 92.76%, which is more than 1.7% higher accuracy than existing methods [22]. Some experimental results can be found in Fig 7. Due to a relatively small number of CCs, our algorithm takes less than 5 seconds on average.

4.4 Application to Skew Estimation

Our method can be applied to a skew estimation problem by modeling text lines as straight lines and computing the average angle of the detected text lines. Although the accuracy of this method is not high compared to conventional methods such as [23], our method is able to handle challenging cases. To be specific, experimental results on 30 vertically flowing text in [23] show that our method achieves an average error of $0.19°$ with a maximum error of $0.5°$, while the method in [23] fails for 3 inputs (See. Table. 4 in [23]).



(a) (b) (c)

Fig. 7. Input and output of our algorithm on the dataset in [3]. (a) Input, (b) Result using $(\epsilon_1, \epsilon_2) = (1.2, 0.9)$, (c) Result using $(\epsilon_1, \epsilon_2) = (10, 10)$. Although the latter setting does not provide text block information, it can provide better text line extraction performance. Therefore, we use the latter setting for the evaluation.

5 Conclusion

In this paper, we have presented a novel approach to document image processing: text block identification and text line extraction. In order to handle complex cases, we assume that the states (line spacing, orientation, and other parameters describing the local properties of text region) of CCs are spatially varying, and the states are estimated using an energy minimization framework. Using the estimated states, we have also presented a new algorithm that performs text block identification and text line extraction. Experimental results on the extensive dataset show that our method is efficient, robust, and provides promising results.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of (NRF) funded by the Ministry of Education, Science and Technology (2009-0083495)

References

1. O'Gorman, L.: The document spectrum for page layout analysis. IEEE Trans. Pattern Anal. Mach. Intell. 15, 1162–1173 (1993)
2. Liang, J., DeMenthon, D., Doermann, D.: Flattening curved documents in images. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2005)
3. Shafait, F., Breuel, T.M.: Document image dewarping contest. In: Int. Workshop on Camera-Based Document Analysis and Recognition, pp. 181–188 (2007)

4. Stamatopoulos, N., Gatos, B., Pratikakis, I., Perantonis, S.: A two-step dewarping of camera document images. In: International Workshop on Document Analysis Systems, pp. 209–216 (2008)
5. Cao, H., Ding, X., Liu, C.: A cylindrical surface model to rectify the bound document. In: International Conference on Computer Vision, ICCV (2003)
6. Koo, H.I., Kim, J., Cho, N.I.: Composition of a dewarped and enhanced document image from two view images. IEEE Trans. Image Process. 18, 1551–1562 (2009)
7. Shafait, F., Keysers, D., Breuel, T.M.: Performance evaluation and benchmarking of six page segmentation algorithms. IEEE Trans. Pattern Anal. Mach. Intell. 30, 941–954 (2008)
8. Zheng, Y., Li, H., Doermann, D.: Machine printed text and handwriting identification in noisy document images. IEEE Trans. Pattern Anal. Mach. Intell. 26, 337–353 (2004)
9. Xiao, Y., Yan, H.: Text region extraction in a document image based on the delaunay tessellation. Pattern Recognition 36, 799–809 (2003)
10. Kise, K., Iwata, M.: Segmentation of page images using the area voronoi diagram. Computer Vision and Image Understanding 70, 370–382 (1998)
11. Bukhari, S.S., Shafait, F., Breuel, T.M.: Coupled snakelet model for curled textline segmentation of camera-captured document images. In: International Conference on Document Analysis and Recognition, pp. 61–65 (2009)
12. Lindeberg, T.: Feature detection with automatic scale selection. International Journal of Computer Vision 30, 79–116 (1998)
13. Yin, F., Liu, C.L.: Handwritten chinese text line segmentation by clustering with distance metric learning. Pattern Recogn. 42, 3146–3157 (2009)
14. de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry. Springer, Heidelberg (2000)
15. Antonacopoulos, A.: Page segmentation using the description of the background. Computer Vision and Image Understanding 70, 350–369 (1998)
16. Bukhari, S., Shafait, F., Breuel, T.: Segmentation of curled textlines using active contours. In: The Eighth IAPR International Workshop on Document Analysis Systems, DAS 2008, pp. 270–277 (2008)
17. Li, Y., Zheng, Y., Doermann, D., Jaeger, S.: Script-independent text line segmentation in freestyle handwritten documents. IEEE Trans. Pattern Anal. Mach. Intell. 30, 1313–1329 (2008)
18. Pilu, M., Pollard, S.: A light-weight text image processing method for handheld embedded cameras. In: BMVC (2002)
19. Pogalin, E., Smeulders, A., Thean, A.: Visual quasi-periodicity. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2008)
20. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. 23, 1222–1239 (2001)
21. Gatos, B., Antonacopoulos, A., Stamatopoulos, N.: Handwriting segmentation contest. In: International Conference on Document Analysis and Recognition, vol. 2, pp. 1284–1288 (2007)
22. Bukhari, S.S., Breuel, T.M., Shafait, F.: Textline information extraction from grayscale camera-captured document images. In: IEEE International Conference on Image Processing (ICIP), pp. 2013–2016 (2009)
23. Dey, P., Noushath, S.: e-pcp: A robust skew detection method for scanned document images. In: Pattern Recognition (2009)

Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding

Huayan Wang[1], Stephen Gould[2], and Daphne Koller[1]

[1] Computer Science Department, Stanford University, CA, USA
[2] Electrical Engineering Department, Stanford Univeristy, CA, USA

Abstract. We address the problem of understanding an indoor scene from a single image in terms of recovering the layouts of the faces (floor, ceiling, walls) and furniture. A major challenge of this task arises from the fact that most indoor scenes are cluttered by furniture and decorations, whose appearances vary drastically across scenes, and can hardly be modeled (or even hand-labeled) consistently. In this paper we tackle this problem by introducing latent variables to account for clutters, so that the observed image is jointly explained by the face and clutter layouts. Model parameters are learned in the maximum margin formulation, which is constrained by extra prior energy terms that define the role of the latent variables. Our approach enables taking into account and inferring indoor clutter layouts *without* hand-labeling of the clutters in the training set. Yet it outperforms the state-of-the-art method of Hedau et al. [4] that requires clutter labels.

1 Introduction

In this paper, we focus on holistic understanding of indoor scenes in terms of recovering the layouts of the major faces (floor, ceiling, walls) and furniture (Fig. 1). The resulting representation could be useful as a strong geometric constraint in a variety of tasks such as object detection and motion planning. Our work is in spirit of recent work on holistic scene understanding, but focuses on indoor scenes.

For parameterizing the global geometry of an indoor scene, we adopt the approach of Hedau et al. [4], which models a room as a *box*. Specifically, given the inferred three vanishing points, we can generate a parametric family of boxes characterizing the layouts of the floor, ceiling and walls. The problem can be formulated as picking the box that best fits the image.

However, a major challenge arises from the fact that most indoor scenes are cluttered by a lot of furniture and decorations. They often obscure the geometric structure of the scene, and also occlude boundaries between walls and the floor. Appearances and layouts of clutters can vary drastically across different indoor scenes, so it is extremely difficult (if not impossible) to model them consistently. Moreover, hand-labeling of the furniture and decorations for training can be an extremely time-consuming (*e.g.*, delineating a chair by hand) and ambiguous task. For example, should windows and the rug be labeled as clutter?

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 435–449, 2010.

Fig. 1. Example results of recovering the "box" (1st row) and clutter layouts (2nd row) for indoor scenes. In the training images we only need to label the "box" but not clutters.

To tackle this problem, we introduce latent variables to represent the layouts of clutters. They are treated as *latent* in that the clutter is not hand-labeled in the training set. Instead, they participate in the model via a rich set of joint features, which tries to explain the observed image by the synergy of the box and the clutter layouts. As we introduce the latent variables we bear in mind that they should account for the *clutter* such as chairs, desks, sofa *etc.* However, the algorithm has no access to any supervision information on the latent variables. Given limited training data, it is hopeless to expect the learning process to figure out the concept of *clutter* by itself. We tackle this problem by introducing *prior* energy terms that capture our knowledge on *what the clutter should be*, and the learning algorithm tries to explain the image by the box and clutter layouts constrained by these prior beliefs. Our approach is attractive that it effectively incorporates complex and structured prior knowledge into a discriminative learning process with little human effort.

We evaluated our approach on the same dataset as used in [4]. Without hand-labeled clutters we achieve the average pixel error rate of 20.1%, in comparison to 26.5% in [4] without hand-labeled clutters, and 21.2% *with* hand-labeled clutters. This improvement can be attributed to three main contributions of our work (1) we introduce latent variables to account for the clutter layouts in a principled manner without hand-labeling them in the training set; (2) we design a rich set of joint features to capture the compatibility between image and the box-clutter layouts; (3) we perform more efficient and accurate inference by making use of the parameterization of the "box" space. The contribution of all of these aspects are validated in our experiments.

1.1 Related Work

Our method is closely related to a recent work of Hedau *et al* [4]. We adopted their idea of modeling the indoor scene geometry by generating "boxes" from

the vanishing points, and using struct-SVM to pick the best box. However, they used supervised classification of surface labels [6] to identify clutters (furniture), and used the trained surface label classifier to iteratively refine the box layout estimation. Specifically, they use the estimated box layout to add features to supervised surface label classification, and use the classification result to lower the weights of "clutter" image regions in estimating the box layout. Thus their method requires the user to carefully delineate the clutters in the training set. In contrast, our latent variable formulation does not require any label of clutters, yet still accounts for them in a principled manner during learning and inference. We also design a richer set of joint feature as well as a more efficient inference method, both of which help boost our performance

Incorporating image context to aid certain vision tasks and to achieve holistic scene understanding have been receiving increasing concern and efforts recently [3,5,6]. Our paper is another work in this direction that focuses on indoor scenes, which demonstrate some unique aspects of due to the geometric and appearance constraints of the room.

Latent variables has been exploited in the computer vision literature in various tasks such as object detection, recognition and segmentation. They can be used to represent visual concepts such as occlusion [11], object parts [2], and image-specific color models [9]. Introducing latent variables into struct-SVM was shown to be effective in several applications [12]. It is also an interesting aspect in our work that latent variables are used in direct correspondence with a concrete visual concept (clutters in the room), and we can visualize the inference result on latent variables via recovered furniture and decorations in the room.

2 Model

We begin by introducing notations to formalize our problem. We use $\boldsymbol{x}$ to denote the input variable, which is an image of an indoor scene; $\boldsymbol{y}$ to denote the output variable, which is the "box" characterizing the major faces (floor, walls, ceiling) of the room; and $\boldsymbol{h}$ to denote the latent variables, which specify the clutter layouts of the scene.

For representing the face layouts variable $\boldsymbol{y}$ we adopt the idea of [4]. Most indoor scenes are characterized by three dominant vanishing points. Given the position of these points, we can generate a parametric family of "boxes". Specifically, taking a similar approach as in [4] we first detect long lines in the image, then find three dominant groups of lines corresponding to three vanishing points. In this paper we omit the details of these preprocessing steps, which can be found in [4] and [8]. As shown in Fig. 2, we compute the average orientation of the lines corresponding to each vanishing point, and name the vanishing point corresponding to mostly horizontal lines as $\boldsymbol{vp}_0$; the one corresponding to mostly vertical lines as $\boldsymbol{vp}_1$; and the other one as $\boldsymbol{vp}_2$.

A candidate "box" specifying the face layouts of the scene can be generated by sending two rays from $\boldsymbol{vp}_0$, two rays from $\boldsymbol{vp}_1$, and connecting the four

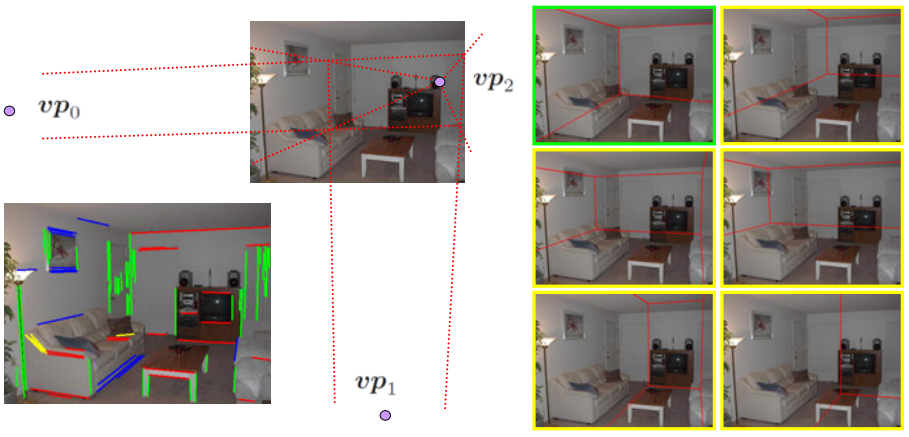


Fig. 2. Lower-Left: We have 3 groups of lines (shown in R, G, B) corresponding to the 3 vanishing points respectively. There are also "o utlier" lines (shown in yellow) which do not belong to any group. **Upper-Left**: A candidate "box" specifying the boundaries between the ceiling, walls and floor is generated. **Right**: Candidate boxes (in yellow frames) generated in this way and the hand-labeled ground truth box layout (in green frame).

intersections with $\boldsymbol{vp}_2$. We use real parameters $\{y_i\}_{i=1}^4$ to specify the position[1] of the four rays sent from $\boldsymbol{vp}_0$ and $\boldsymbol{vp}_1$. Thus the position of the vanishing points and the value of $\{y_i\}_{i=1}^4$ completely determine a box hypothesis assigning each pixel a face label, which has five possible values {*ceiling*, *left-wall*, *right-wall*, *front-wall*, *floor*}. Note that some of the face labels could be absent; for example one might only observe *right-wall*, *front-wall* and *floor* in an image. In that case, some value of y_i would give rise to a ray that does not intersect with the extent of the image. Therefore we can represent the output variable $\boldsymbol{y}$ by only 4 dimensions $\{y_i\}_{i=1}^4$ thanks to the strong geometric constraint of the vanishing points[2]. One can also think of $\boldsymbol{y}$ as the face labels for all pixels. We also define a base distribution $p_0(\boldsymbol{y})$ over the output space estimated by fitting a multivariate Gaussian with diagonal covariance via maximum likelihood to the label boxes in the training set. The base distribution is used in our inference method.

To compactly represent the clutter layout variable $\boldsymbol{h}$, we first compute an over-segmentation of the image using mean-shift [1]. Each image is segmented into a number (typically less than a hundred) of regions, and for each region we assign it to either *clutter* or *non-clutter*. Thus the latent variable $\boldsymbol{h}$ is a binary

[1] There could be different design choices for parameterizing the "position" of a ray sent from a vanishing point. We use the position of its intersection with the image central line (use vertical and horizontal central line for $\boldsymbol{vp}_0$ and $\boldsymbol{vp}_1$ respectively).

[2] Note that $\boldsymbol{y}$ resides in a confined domain. For example, given the prior knowledge that the camera cannot be above the ceiling or beneath the floor, the two rays sent by $\boldsymbol{vp}_0$ must be on different sides of $\boldsymbol{vp}_2$. Similar constraints also apply to $\boldsymbol{vp}_1$.

vector with the same dimensionality as the number of regions in the image that resulted from the over-segmentation.

We now define the energy function $\mathbf{E}_{\boldsymbol{w}}$ that relates the image, the box and the clutter layouts:

$$\mathbf{E}_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}) = \langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}) \rangle - \mathbf{E}^0(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}) \,. \tag{1}$$

$\boldsymbol{\Psi}$ is a joint feature mapping that contains a rich set of features measuring the compatibility between the observed image and the box-clutter layouts, taking into account image cues from various aspects including color, texture, perspective consistency, and overall layout. $\boldsymbol{w}$ contains the weights for the features that needs to be learned. $\mathbf{E}^0$ is an energy term that captures our prior knowledge on the role of the latent variables. Specifically, it measures the appearance consistency of the major faces (floor and walls) when the clutters are taken out, and also takes into account the overall clutterness of each face. Intuitively, it defines the latent variables (clutter) to be *things that appears inconsistently in each of the major faces*. Details about $\boldsymbol{\Psi}$ and $\mathbf{E}^0$ are introduced in Section 3.3.

The problem of recovering the face and clutter layouts can be formulated as:

$$(\bar{\boldsymbol{y}}, \bar{\boldsymbol{h}}) = \arg\max_{(\boldsymbol{y}, \boldsymbol{h})} \mathbf{E}_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}) \,. \tag{2}$$

3 Learning and Inference

3.1 Learning

Given the training set $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^m$ with hand-labeled box layouts, we learn the parameters $\boldsymbol{w}$ discriminatively by adapting the large margin formulation of struct-SVM [10,12],

$$\min_{\boldsymbol{w}, \boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{m}\sum_{i=1}^{m} \xi_i \,, \quad \text{s.t. } \forall i, \ \xi_i \geq 0 \quad \text{and} \tag{3}$$

$$\forall i, \ \boldsymbol{y} \neq \boldsymbol{y}_i, \quad \max_{\boldsymbol{h}_i} \mathbf{E}_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{h}_i) - \max_{\boldsymbol{h}} \mathbf{E}_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{y}, \boldsymbol{h}) \geq 1 - \frac{\xi_i}{\Delta(\boldsymbol{y}, \boldsymbol{y}_i)} \,, \tag{4}$$

where $\Delta(\boldsymbol{y}, \boldsymbol{y}_i)$ is the loss function that measures the difference between the candidate output $\boldsymbol{y}$ and the ground truth $\boldsymbol{y}_i$. We use pixel error rate (the percentage of pixels that are labeled differently by the two box layouts) as the loss function.

As $\mathbf{E}^0$ encodes the prior knowledge, it is fixed to constrain the learning process of model parameters $\boldsymbol{w}$. Without the slack variables ξ_i the constraints (4) essentially state that, for each training image i, any candidate box layout $\widehat{\boldsymbol{y}}$ cannot better explain the image than the ground truth layout $\boldsymbol{y}_i$. Maximizing the compatibility function over the latent variables gives the clutter layouts that best explain the image and box layouts under the current model parameters. Since the model can never fully explain the intrinsic complexity of real-world images,

we have to slacken the constraints by the slack variables, which are scaled by the loss function $\Delta(\widehat{\boldsymbol{y}}, \boldsymbol{y}_i)$ indicating that hypothesis deviates more from the ground truth violating the constraint would incur a larger penalty.

The learning problem is difficult because the number of constraints in (4) is infinite. Even if we discretize the parameter space of $\boldsymbol{y}$ in some way, the total number of constraints is still huge. And each constraint involves an embedded inference problem for the latent variables. Generally this is tackled by gradually adding most violated constraints to the optimization problem [7,10], which involves an essential step of *loss augmented inference* that tries to find the output variable $\hat{\boldsymbol{y}}$ for which the constraint is most violated given the current parameters $\boldsymbol{w}$. In our problem, it corresponds to following inference problem:

$$(\widehat{\boldsymbol{y}}, \widehat{\boldsymbol{h}}) = \arg\max_{\boldsymbol{y},\boldsymbol{h}} (1 + \mathbf{E}_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{y}, \boldsymbol{h}) - \mathbf{E}_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{h}_i)) \cdot \Delta(\boldsymbol{y}, \boldsymbol{y}_i) \,, \tag{5}$$

where the latent variables $\boldsymbol{h}_i$ should take the value that best explains the ground truth box layout under current model parameters:

$$\boldsymbol{h}_i = \arg\max_{\boldsymbol{h}} \mathbf{E}_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{h}) \,. \tag{6}$$

The overall learning algorithm (follows from [10]) is shown in Algorithm 1. In the rest of this section, we will elaborate on the inference problems of (5) and (6), as well as the details of $\boldsymbol{\Psi}$ and $\mathbf{E}^0$.

Algorithm 1. Overall Learning Procedure

Input: $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^m$, C, ϵ_{final}
Output: $\boldsymbol{w}$
$Cons \leftarrow \emptyset$
$\epsilon \leftarrow \epsilon_0$
repeat
  for $i = 1$ to m **do**
    find $(\widehat{\boldsymbol{y}}, \widehat{\boldsymbol{h}})$ by solving (5) using Algorithm 2
    if the constraint in (4) corresponding to $(\widehat{\boldsymbol{y}}, \widehat{\boldsymbol{h}})$ is violated more than ϵ **then**
      add the constraint to $Cons$
    end if
  end for
  update $\boldsymbol{w}$ by solving the QP given $Cons$
  for $i = 1$ to m **do**
    update $\boldsymbol{h}_i$ by solving (6)
  end for
  if # new constraints in last iteration is less than threshold **then**
    $\epsilon \leftarrow \epsilon/2$
  end if
until $\epsilon < \epsilon_{final}$ and # new constraints in last iteration is less than threshold

3.2 Approximate Inference

Because the joint feature mapping $\boldsymbol{\Psi}$ and prior energy $\mathbf{E}^0$ are defined in a rather complex way in order to take into account various kinds of image cues, the inference problems (2), (5) and (6) cannot be solved analytically. In [4] there was no latent variable $\boldsymbol{h}$, and the space of $\boldsymbol{y}$ is still tractable for simple discretization, so the constraints for struct-SVM can be pre-computed for each training image before the main learning procedure. However in our problem we are confronting the combinatorial complexity of $\boldsymbol{y}$ and $\boldsymbol{h}$, which makes it impossible to pre-compute all constraints.

For inferring $\boldsymbol{h}$ given $\boldsymbol{y}$, we use iterated conditional modes (ICM) [13]. Namely, we iteratively visit all segments, and flip a segment (between *clutter* and *non-clutter*) if it increase the objective value, and we stop the process if no segment is flipped in last iteration. To avoid local optima we start from multiple random initializations. For inferring both $\boldsymbol{y}$ and $\boldsymbol{h}$, we use stochastic hill climbing for $\boldsymbol{y}$, and the algorithm is shown in Algorithm 2.

The test-time inference procedure (2) is handle similarly as the loss augmented inference (5) but with a different objective. We can use a looser convergence criterion for (5) to speed up the process as it has to be performed multiple times in learning. The overall inference process is shown in Algorithm 2.

Algorithm 2. Stochastic Hill-Climbing for Inference

Input: $\boldsymbol{w}$, $\boldsymbol{x}$
Output: $\bar{\boldsymbol{y}}$, $\bar{\boldsymbol{h}}$
for a number of random seeds **do**
  sample $\bar{\boldsymbol{y}}$ from $p_0(\boldsymbol{y})$
  $\bar{\boldsymbol{h}} \leftarrow \arg\max_{\boldsymbol{h}} \mathbf{E}_{\boldsymbol{w}}(\boldsymbol{x}, \bar{\boldsymbol{y}}, \boldsymbol{h})$ by ICM
  repeat
    repeat
      perturb a parameter of $\boldsymbol{y}$ as long as it increases the objective
    until convergence
    $\bar{\boldsymbol{h}} \leftarrow \arg\max_{\boldsymbol{h}} \mathbf{E}_{\boldsymbol{w}}(\boldsymbol{x}, \bar{\boldsymbol{y}}, \boldsymbol{h})$ by ICM
  until convergence
end for

In experiments we also compare to another inference method that does not make use of the continuous parameterization of $\boldsymbol{y}$. Specifically we independently generate a large number of candidate boxes from $p_0(\boldsymbol{y})$, infer the latent variable for each of them, and pick the one with the largest objective value. This is similar to the inference method used in [4], in which they independently evaluate all hypothesis boxes generated from a uniform discretization of the output space.

3.3 Priors and Features

For making use of color and texture information, we assign a 21 dimensional appearance vector to each pixel, including HSV values (3), RGB values (3),

Gaussian filter in 3 scales on all 3 Lab color channels (9), Sobel filter in 2 directions and 2 scales (4), and Laplacian filter in 2 scales (2). Each dimension is normalized for each image to have zero mean and unit variance.

The prior energy-term $\mathbf{E}^0$ consists of 2 parts,

$$\mathbf{E}^0(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}) = \alpha^a \mathbf{E}^a(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}) + \alpha^c \mathbf{E}^c(\boldsymbol{y}, \boldsymbol{h}) \,. \tag{7}$$

The first term $\mathbf{E}^a$ summarizes the **a**ppearance variance of each major face excluding all clutter segments, which essentially encodes the prior belief that the major faces should have a relatively consistent appearance after the clutters are taken out. Specifically $\mathbf{E}^a$ is computed as the variance of the appearance value within a major face excluding clutter, summed over all the 21 dimensions of appearance values and 5 major faces. The second term $\mathbf{E}^c$ penalizes **c**lutterness of the scene to avoid taking out almost everything and leaving a tiny uniform piece that is very consistency in appearance. Specifically, for each face we compute $\exp(\beta s)$, where s is the area percentage of clutter in that face and β is a constant factor. This value is then averaged over the 5 faces weighted by their areas. The reason for adopting the exponential form is that it demonstrates superlinear penalty as the percentage of clutter increases. The relative weights between these 2 terms as well as the constant factor β were determined by cross-validation on the training set and then fixed in the learning process.

The features in $\boldsymbol{\Psi}$ come from various aspects of image cues as summarized below (228 features in total).

1. **Face Boundaries:** Ideally the boundaries between the 5 major faces should either be explained by a long line or occluded by some furniture. Therefore we introduce 2 features for each of the 8 boundaries[3], computed by the percentage of its length that is (1) in a clutter segment and (2) approximately overlapping with a line. So there are 16 features in this category.
2. **Perspective consistency:** The idea behind perspective consistency features is adopted from [4]. The lines in the image can be assigned into 3 groups corresponding to the 3 vanishing points (Fig. 2). For each major face, we are more likely to observe lines from 2 of the 3 groups. For example, on the front wall we are more likely to observe lines belonging to $\boldsymbol{vp}_0$ and $\boldsymbol{vp}_1$, but not $\boldsymbol{vp}_2$. In [4] they defined 5 features by computing the length percentage of lines from the "correct" groups for each face. In our work we enlarge the number of features to leave the learning algorithm with more flexibility. Specifically we count the total length of lines from all 3 groups in all 5 faces, and treating clutter and non-clutter segments separately, which results in $3 \times 5 \times 2 = 30$ features in this category.
3. **Cross-face difference:** For the 21 appearance values, we compute the difference between the 8 pairs of adjacent faces (excluding clutters), which results in 168 features.

[3] If all 5 faces are present, there are 8 boundaries between them.

4. **Overall layouts:** For each of 5 major faces, we use a binary feature indicating whether it is observable or not, and we also use a real feature for its area percentage in the image. Finally, we compute the likelihood of each of the 4 parameters $\{y_i\}_{i=1}^4$ under $p_0(\boldsymbol{y})$. So there are 14 features in this category.

4 Experimental Results

For experiments we use the same dataset[4] as used in [4]. The dataset consists of 314 images, and each image has hand-labeled box and clutter layouts. They also provided the training-test split (209 for training, 105 for test) on which they reported results in [4]. For comparison we use the same training-test split and achieve a pixel-error-rate of 20.1% *without* clutter labels, comparing to 26.5% in [4] without clutter labels and 21.2% with clutter labels. Detailed comparisons are shown in Table 1 (the last four columns are explained in the following subsections).

Table 1. Quantitative results. **Row 1:** pixel error rate. **Row 2 & 3**: the number of test images (out of 105) with pixel error rate under 20% & 10%. **Column 1 ([6])**: Hoiem et al.'s region labeling algorithm. **Column 2 ([4] w/o)**: Hedau et al.'s method without clutter label. **Column 3 ([4] w/)** : Hedau et al.'s method with clutter label (iteratively refined by supervised surface label classification [6]). The first 3 columns are directly copied from [4]. **Column 4 (Ours w/o)**: Our method (without clutter label). **Column 5 (w/o prior)**: Our method without the prior knowledge constraint. **Column 6 ($\boldsymbol{h} = \mathbf{0}$)**: Our method with latent variables fixed to be zeros (assuming "no clutter"). **Column 7 ($\boldsymbol{h} =$ GT)**: Our method with latent variables fixed to be hand-labeled clutters in learning. **Column 8 (UB)**: Our method with latent variables fixed to be hand-labeled clutters in both learning and inference. In this case the testing phase is actually "cheating" by making use of the hand-labeled clutters, so the results can only be regarded as some upperbound. The deviations in the results are due to the randomization in both learning and inference. They are estimated over multiple runs of the entire procedure.

	[6]	[4] w/o	[4] w/	Ours w/o	w/o prior	$\boldsymbol{h} = \mathbf{0}$	$\boldsymbol{h} =$ GT	UB
Pixel	28.9%	26.5%	21.2%	**20.1±0.5%**	21.5±0.7%	22.2±0.4%	24.9±0.5%	19.2±0.6%
≤20%	–	–	–	**62±3**	58±4	57±3	46±3	67±3
≤10%	–	–	–	**30±3**	24±2	25±3	20±2	37±4

In order to validate the effects of prior knowledge in constraining the learning process, we take out the prior knowledge by adding the two terms $\mathbf{E}^a$ and $\mathbf{E}^c$ as ordinary features and try to learn their weights. The performance of recovering

[4] The dataset is available at
https://netfiles.uiuc.edu/vhedau2/www/groundtruth.zip

Fig. 3. Sample results for comparing learning with and without prior constraints. The 1st and 2nd column are the result of learning with prior constraints. The 3rd and 4th column are the result of learning without prior constraints. The clutter layouts are shown by removing all non-clutter segments. In many cases recovering more reasonable clutters does help in recovering the correct box layout.

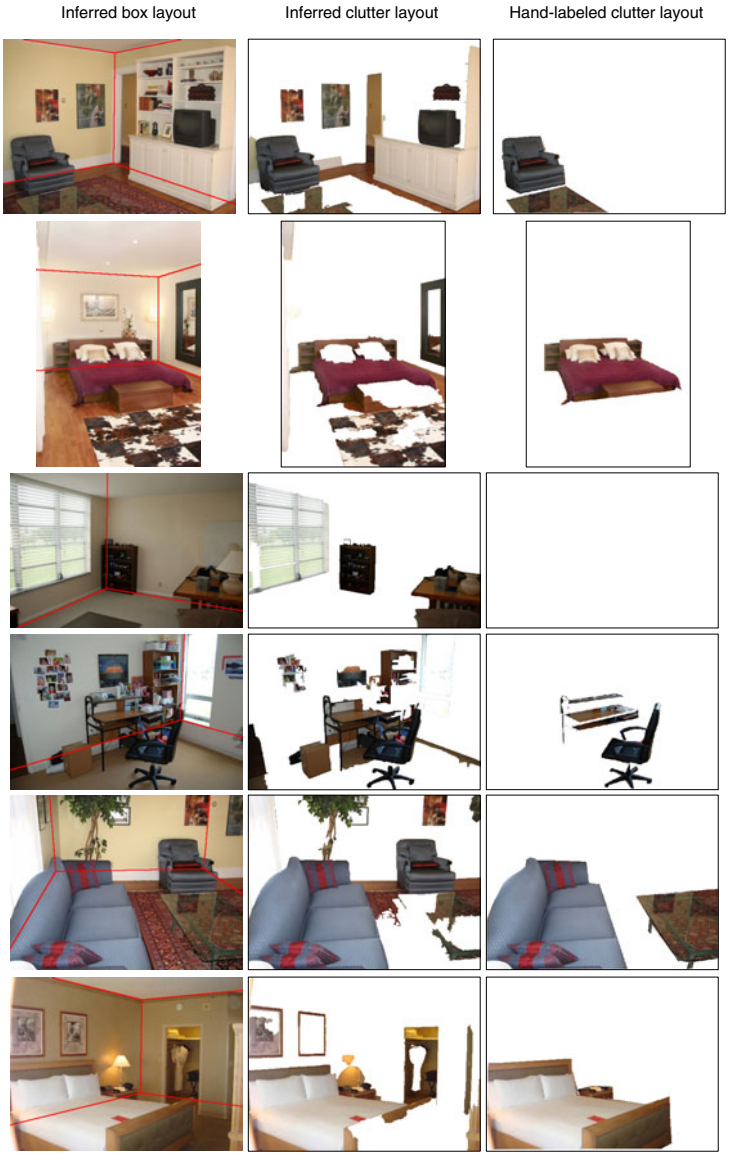


Fig. 4. Sample results for comparing the recovered clutters by our method and the hand-labeled clutters in the dataset. The 1st and 2nd column are recovered box and clutter layouts by our method. The 3rd column (right) is the hand-labeled clutter layouts. Our method usually recovers more objects as "clutter" than people would bother to delineate by hand. For example, the rug with a different appearance from the floor in the 2nd image, paintings on the wall in the 1st, 4th, 5th, 6th image, and the tree in the 5th image. There are also major pieces of furniture that are missing in the hand-labels but recovered by our method, such as the cabinet and TV in the 1st image, everything in the 3rd image, and the small sofa in the 5th image.

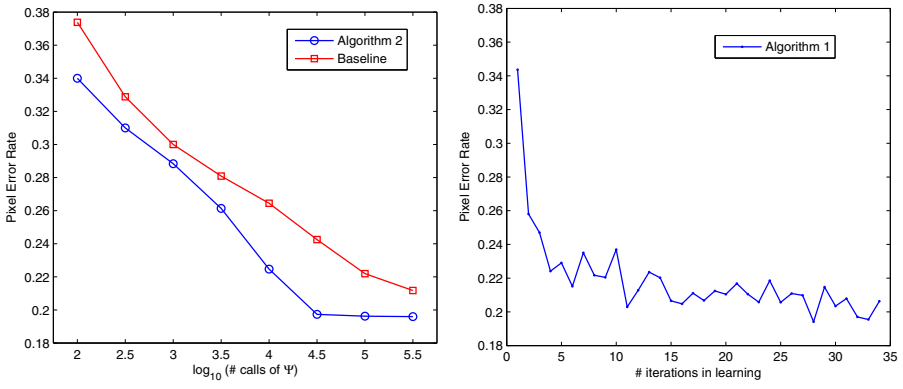


Fig. 5. Left: Comparison between the inference method described in Algorithm 2 and the baseline inference method that evaluates hypotheses independently. **Right:** Empirical convergence evaluation for the learning procedure.

box layouts in this case is shown in Table 1, column 5. Although the difference between column 4 and 5 (Table 1) is small, there are many cases where recovering more reasonable clutters does help in recovering the correct box-layout. Some examples are shown in Figure 3, where the 1st and 2nd column (from left) are the box and clutter layouts recovered by the learned model with prior constraints, and the 3rd and 4th column are the result of learning without prior constraints. For example, in the case of the 3rd row (Fig. 3), the boundary between the *floor* and the *front-wall* (the wall on the right) is correctly recovered even though it is largely occluded by the bed, which is correctly inferred as "clutter", and the boundary is probably found by the appearance difference between the floor and the wall. However, with the model learned without prior constraints, the bed is regarded as non-clutter whereas the major parts of the floor and walls are inferred as clutter (this is probably because the term $\mathbf{E}^c$ is not acting effectively with the learned weights), so it appears that the boundary between the *floor* and the *front-wall* is decided incorrectly by the difference between the white pillow and blue sheet.

We tried to fix the latent variables $\boldsymbol{h}$ to be all zeros. The results are shown in column 6 of Table 1. Note that in obtaining the result of 26.5% without clutter labels in [4], they only used "perspective consistency" features, although other kinds of features are incorporated as they resort to the clutter labels and the supervised surface label classification method in [6]. By fixing $\boldsymbol{h}$ to be all zeros (assuming no clutter) we actually decomposed our performance improvement upon [4] into two parts: (1) using the richer set of features, and (2) accounting for clutters with latent variables. Although the improvement brought by the richer set of features is larger, the effect of accounting for clutters is also significant.

We also tried fix the latent variables $\boldsymbol{h}$ to be the hand-labeled clutter layouts[5]. The results are shown in column 7 of Table 1. We quantitatively compared our recovered clutter to the hand-labeled clutters, and the average pixel difference is around 30% on both the training and test set. However this value does not necessarily reflect the quality of our recovered clutters. In order to justify this, we show some comparisons between the hand-labeled clutters and the recovered clutters (from the test set) by our method in Fig. 4. Generally the hand labels include much less clutters than our algorithm recovers. Because delineating objects by hand is very time consuming, usually only one or two pieces of major furniture are labeled as clutter. Some salient clutters are missing in the hand-labels such as the cabinet and the TV in the image of the 1st row (Fig. 4), the smaller sofa in the image of the 5th row, and nothing is labeled in the image of the 3rd row. Therefore it is not surprising that learning with the hand-labeled clutter does not resulting in a better model (Table 1, column 7). Additionally, we also tried to fix the latent variable to be the hand-labeled clutters in *both* learning and inference. Note that the algorithm is actually "cheating" as it has access to the labeled clutters even in the testing phase. In this case it does give slightly better results (Table 1, column 8) than our method.

Although our method has improved the state-of-the-art performance on the dataset, there are still many cases where the performance is not satisfiable. For example in the 3rd image of Fig. 4, the ceiling is not recovered even though there are obvious image cues for it, and in the 4th-6th image of Fig. 4, the boundaries between the floor and the wall are not estimated accurately. There is around 6-7% (out of the 20.1%) of the pixel error due to incorrect vanishing point detection results[6].

We compare our inference method (Algorithm 2) to the baseline method (evaluating hypotheses independently) described in Section 3.2. Fig. 5 (Left) shows the average pixel error rate over test set versus the number of calls to the joint feature mapping $\boldsymbol{\Psi}$ in log scale, which could be viewed as a measure of running time. The difference between the two curves is actually huge as we are plotting in log-scale. For example, for reaching the same error rate of 0.22 the baseline method would take roughly 10 times more calls to $\boldsymbol{\Psi}$.

As we have introduced many approximations into the learning procedure of latent struct-SVM, it is hard to theoretically guarantee the convergence of the learning algorithm. In Fig. 5 (Right) we show the performance of the learned model on test set versus the number of iterations in learning. Empirically the learning procedure approximately converges in a small number of iterations,

[5] The hand-labeled clutters in the dataset are not completely compatible with our over-segmentation, *i.e.*, some segments may be partly labeled as clutter. In that case, we assign 1 to a binary latent variable if over 50% of the corresponding segment is labeled as clutter. The pixel difference brought by this "approximation" is 3.5% over the entire dataset, which should not significantly affect the learning results.

[6] The error rate of 6-7% is estimated by assuming a perfect model that always picks the best box generated from the vanishing point detection result, and performing stochastic hill-climbing to infer the box using the perfect model.

although we do observe some fluctuation due to the randomized approximation used in the loss augmented inference step of learning.

5 Conclusion

In this paper we addressed the problem of recovering the geometric structure as well as clutter layouts from a single image. We used latent variables to account for indoor clutters, and introduced prior terms to define the role of latent variables and constrain the learning process. The box and clutter layouts recovered by our method can be used as a geometric constraint for subsequent tasks such as object detection and motion planning. For example, the box layout suggests relative depth information, which constrains the scale of the objects we would expect to detect in the scene.

Our method (without clutter labels) outperforms the state-of-the-art method (with clutter labels) in recovering the box layout on the same dataset. And we are also able to recover the clutter layouts *without* hand-labeling of them in the training set.

Acknowledgements

This work was supported by the National Science Foundation under Grant No. RI-0917151, the Office of Naval Research under the MURI program (N000140710747) and the Boeing Corporation.

References

1. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Transactions on PAMI 24(5) (2002)
2. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE Transactions on PAMI (to appear, 2010)
3. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV (2009)
4. Hedau, V., Hoiem, D., Forsyth, D.: Recovering the spatial layout of cluttered room. In: ICCV 2009 (2009)
5. 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)
6. Hoiem, D., Efros, A., Hebert, M.: Recovering surface layout from an image. IJCV 75(1) (2007)
7. Joachims, T., Finley, T., Yu, C.-N.: Cutting-Plane Training of Structural SVMs. Machine Learning 77(1), 27–59 (2009)
8. Rother, C.: A new approach to vanishing point detection in architectural environments. IVC 20 (2002)
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. IJCV (2007)

10. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y., Singer, Y.: Large margin methods for structured and interdependent output variables. JMLR 6, 1453–1484 (2005)
11. Vedaldi, A., Zisserman, A.: Structured output regression for detection with partial occlusion. In: NIPS (2009)
12. Yu, C.-N., Joachims, T.: Learning structural SVMs with latent variable. In: ICML (2009)
13. Besag, J.: On the statistical analysis of dirty pictures (with discussions). Journal of the Royal Statistical Society, Series B 48, 259–302 (1986)

Simultaneous Segmentation and Figure/Ground Organization Using Angular Embedding

Michael Maire

California Institute of Technology - Pasadena, CA, 91125
mmaire@caltech.edu

Abstract. Image segmentation and figure/ground organization are fundamental steps in visual perception. This paper introduces an algorithm that couples these tasks together in a single grouping framework driven by low-level image cues. By encoding both affinity and ordering preferences in a common representation and solving an Angular Embedding problem, we allow segmentation cues to influence figure/ground assignment and figure/ground cues to influence segmentation. Results are comparable to state-of-the-art automatic image segmentation systems, while additionally providing a global figure/ground ordering on regions.

1 Introduction

Segmentation, the task of partitioning an image into homogeneous regions, and figure/ground organization, the task of assigning ownership of a contour to one of the two regions it separates, are both active and open problems in computer vision. Historically, more attention has been paid to segmentation, though some important studies of figure/ground exist, focusing on contour and junction structure [13,11,25,32] or specific cues [10] such as convexity [21] or lower-region [29]. Recent work has revived interest on figure/ground discrimination [24,16] and the related problem of depth ordering [15,26].

Previous work starts from the assumption that figure/ground organization occurs after contours [24] or regions [16] have been obtained and designs algorithms that require an image segmentation as input. Hoiem *et al.* [15] fix an initial oversegmentation and iterate region-merging and depth estimation steps. It is not yet known where figure/ground discrimination occurs in biological visual systems [22], with, as noted by Ren *et al.* [24], some evidence for early availability of a contour ownership signal [34].

Most automatic image segmentation algorithms ignore figure/ground organization, producing a two-dimensional partition of the image with no notion of figure or depth ordering [28,6,30,8,27,1,23,3]. Other work treats depth recovery itself as an end goal, exploiting segmentation along with scene geometry (*e.g.* estimated horizon location) or object knowledge to help build a three-dimensional rendering of the image. Examples include the photo pop-up work of Hoiem *et al.* [14] and the scene reconstruction work of Gould *et al.* [12].

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 450–464, 2010.

This paper takes a different approach, attempting to bring figure/ground cues into perceptual processing as early as possible. We want to build a generic segmentation and figure/ground reasoning stage with the goal of enriching the image representation available to tasks such as object recognition.

We accomplish this by extending a leading image segmentation method based on spectral partitioning into an algorithm that recovers figure/ground organization as well. The system we extend is that of Arbeláez *et al.* [3], which currently provides the best performance of all automatic segmentation algorithms across a range of benchmarks on the Berkeley Segmentation Dataset (BSDS) [19,18]. Our key insight is to replace their core grouping machinery, based on Normalized Cuts [28] and described in Maire *et al.* [17], with the more general Angular Embedding of Yu [31]. Angular Embedding allows us to represent both segmentation and figure/ground relations and solve for both at once.

To our knowledge, no previous work recovers segmentation and figure/ground for natural images in a single step. Yu and Shi [33] attempt to use pairwise repulsion cues to fuse figure/ground with segmentation in spectral graph theory. However, they show only one example on T-junctions. A core component of our solution, Angular Embedding, was only recently introduced [31] and we believe our work is the first application of this technique to non-synthetic images.

The most closely related work to ours is that of Ren *et al.* [24] and Leichter and Lindenbaum [16], both of which focus on solving an easier problem than the one suggested here. Leichter and Lindenbaum take the human-drawn ground-truth segmentations [19] and human figure/ground annotations [10] of the BSDS images and learn a conditional random field (CRF) for assigning boundary ownership. They use curve and junction potentials, exploiting convexity, lower-region, fold/cut, and parallelism cues. Their impressive results of 82.8% correct figure/ground assignments (chance being 50%) are only obtained when *testing on human-drawn ground-truth segmentations*. Testing on automatically generated curves, they obtain only 69.1% accuracy, similar to the 68.9% accuracy reported by Ren *et al.* [24] on automatically generated contours. In contrast to our integrated approach, both works cast figure/ground assignment as a step to be run after first solving for a segmentation.

The notion of figure/ground used in this work is that used by Ren *et al.* [24]. Namely, a figural region is defined according to human perception. We simply attempt to replicate human behavior by training on human-annotated data. As previously pointed out [20,16], this means that figure/ground ordering does not necessarily correspond to depth or occlusion ordering. For example, humans may indicate strong figure/ground percepts due to markings on flat surfaces.

Our choice to follow this convention for the meaning of figural regions is consistent with the goal of targeting our output for use in perceptual tasks such as recognition rather than geometric scene reconstruction. It is also partially motivated by convenience, as it allows us to train our figure/ground classifier on the same dataset, the BSDS, as our segmentation algorithm, due to the availability of pre-existing annotations [19,10]. Consequently, our work is not directly comparable to that for which depth ordering or three-dimensional reconstruction is the

ultimate goal [15,26]. However, since our algorithm for the combined segmentation and figure/ground problem is agnostic to the source of the local figure/ground cues, it is conceivable that future work could re-purpose our system to solve a depth ordering problem.

Section 2 describes our new grouping framework for simultaneous segmentation and figure/ground assignment. It is compatible with any appropriate sources of pairwise similarity and ordering cues. Section 3 details our particular choice for the local ordering cues. Section 4 presents both qualitative and quantitative results for fully automatic segmentation and figure/ground organization. Our system compares favorably to others on the segmentation task, while producing a global figural ranking of regions at minimal additional computational cost.

2 Adding Ordering to Segmentation

Figure 1 outlines our algorithm for simultaneous segmentation and figure/ground organization. We extend previous work on segmentation alone [17,3] to incorporate figure/ground information through the use of Angular Embedding [31] as a globalization procedure. Removing the vertical pathway for figure/ground information shown on the right side of Figure 1 and replacing Angular Embedding by Normalized Cuts, one recovers the segmentation-only pipeline of Arbeláez *et al.* [3]. To make this paper as self-contained as possible, we briefly review the core portions of the relevant previous work, before describing how to bring in figure/ground cues in the form of pairwise ordering preferences.

2.1 Spectral Partitioning

Spectral clustering, and specifically Normalized Cuts [28], have long been popular techniques for image segmentation. Recently, [17] achieve excellent results by using Normalized Cuts in a "soft" manner as a globalization stage for contour detection. The approach taken is to define a sparse affinity matrix connecting nearby pixels p and q with weight determined by the *intervening contour* [9] cue:

$$W(p,q) = \exp\left(-\max_{(x,y)\in\overline{pq}}\{mPb(x,y)\}/\rho\right) \tag{1}$$

where $\overline{pq}$ is the line segment connecting p and q, ρ is a constant, and mPb stands for multiscale *probability of boundary* [18,17] and measures the probability that the pixel at location (x,y) lies on a boundary contour. A classifier trained using local brightness, color, and texture cues predicts mPb at each image location.

To obtain global contour strength from these local measurements, one forms matrix D whose diagonal contains the row-sums of W and solves for the generalized eigenvectors $\{v_0, v_1, ..., v_n\}$ of the system:

$$(D - W)v = \lambda Dv \tag{2}$$

corresponding to the $n+1$ smallest eigenvalues $0 = \lambda_0 \leq \lambda_1 \leq ... \leq \lambda_n$. Associating with each pixel p the length n descriptor containing the p^{th} entry from

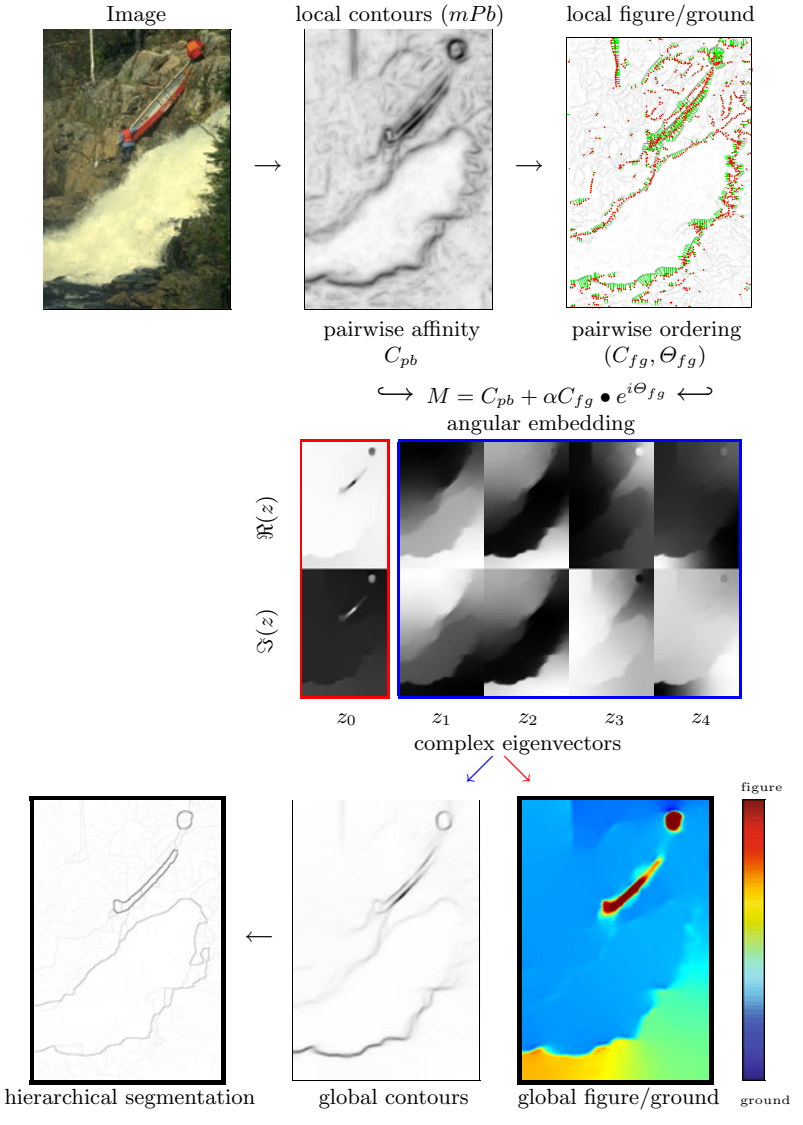


Fig. 1. Segmentation and figure/ground organization. From the image (*top left*) we compute the probability of boundary (*pb*) [18] using the multiscale detector (*mPb*) of [17] (*top middle*). Nonmax-suppressed *mPb* contours are fed to a local shape-based figure/ground classifier (*top right*), whose output is shown by green vectors with red tips drawn towards the predicted figural side. The *mPb* signal defines a pairwise affinity between neighboring pixels via intervening contour [9]. The figure/ground classifier defines a longer-range pairwise ordering. A generalized affinity matrix M captures both sources of information. Solving an Angular Embedding [31] problem yields complex eigenvectors (*middle*) which encode both segmentation (*bottom left*) and global figure/ground ordering (*bottom right*). Red indicates more figural regions.

each eigenvector creates an embedding in $\mathbb{R}^n$. Equivalently, $\{v_1, ..., v_n\}$ can be viewed as a stack of n images for which the segmentation problem is now easy. Convolving with Gaussian directional derivative filters produces a robust measure of contour strength. Applying tools from image morphology then permits construction of a hierarchical segmentation from these high-quality contours [3].

For an intuition behind this machinery, note that the first eigenvector v_1 is the exact global minimizer of the following error measure [28]:

$$\inf_{v^T D 1 = 0} \frac{\sum_p \sum_q W(p,q)(v(p) - v(q))^2}{\sum_p D(p)(v(p))^2} \tag{3}$$

The weight on the squared difference forces the eigenvector to take similar values for pixels with high affinity.

2.2 Angular Embedding

The spectral partitioning algorithm of the previous section produces real-valued eigenvectors. Angular Embedding [31] is an alternative technique that produces complex-valued eigenvectors. Our problem is no longer defined by the symmetric real-valued matrix W, but instead by a pair of real-valued matrices (C, Θ), where C is a symmetric *confidence* matrix analogous to W, and Θ is a skew-symmetric *ordering* matrix. The goal is to produce an embedding into the unit circle in the complex plane such that sorting the resulting points by their angle respects the pairwise local ordering constraints defined by Θ. Confidence matrix C encodes the relative importance of each constraint.

Specifically, let $z(p) \in \mathbb{C}$ denote the embedding of p. We minimize error:

$$\varepsilon = \sum_p D(p) \cdot |z(p) - \tilde{z}(p)|^2 \tag{4}$$

where D is again a diagonal degree matrix with:

$$D(p) = \frac{\sum_q C(p,q)}{\sum_{p,q} C(p,q)} \tag{5}$$

and $\tilde{z}(p)$ is the position of p estimated from its neighbors through a rotation by their relative ordering:

$$\tilde{z}(p) = \sum_q \tilde{C}(p,q) \cdot e^{i\Theta(p,q)} \cdot z(q) \tag{6}$$

$$\tilde{C}(p,q) = \frac{C(p,q)}{\sum_q C(p,q)} \tag{7}$$

$|z - \tilde{z}|$ is an appropriate error measure as z and $\tilde{z}$ coincide if and only if the embedding perfectly fulfills all local orderings with positive confidence [31].

Rewriting the above in matrix form requires one to minimize:

$$\varepsilon = z^* W z \tag{8}$$

subject to $z = e^{i\theta}$ for a real-valued vector θ where:

$$W = (I - M)^* D (I - M) \tag{9}$$

$$M = Diag(C1)^{-1} C \bullet e^{i\Theta} \tag{10}$$

$$D = Diag(C1 \cdot (1^* C1)^{-1}) \tag{11}$$

and * denotes complex conjugate transpose, $\bullet$ is the matrix Hadamard product, I is the identity matrix, 1 is a column vector of ones, $Diag(\cdot)$ is a matrix with its vector argument on the main diagonal, $i = \sqrt{-1}$ and exponentiation acts element-wise. Relaxing the constraint that z lie on the unit circle to $z^* D z = 1$ yields the solution as the angle of the first eigenvector, $\angle z_0$, of the generalized eigenproblem specified by (W, D). Unlike (2), for nontrivial Θ, we have $\lambda_0 \neq 0$ and all of the eigenvectors, including z_0, are meaningful.

2.3 Short-Range Attraction, Long-Range Ordering

We use the additional expressive freedom of Angular Embedding to encode both pairwise segmentation cues and pairwise figure/ground cues in the common representation defined by (C, Θ). Let us now write the affinity matrix defined by intervening contour (1) as $C_{pb}(p, q)$. It uses the probability of boundary (pb) cue to place a confidence on the event that pixels p and q lie in the same region. This cue yields no information on relative figural ordering, so we set $\Theta_{pb}(p, q) = 0 \ \forall p, q$.

Suppose we also have a classifier $f(x, y) \rightarrow [-1, 1]$ that, at an edge pixel (x, y) lying on a contour obtained by nonmax-suppression [5] of mPb, predicts which side of the edge is figural. Let p and q be the pixels located a fixed distance r from (x, y), on opposite sides (left and right, respectively) of the edge, in the direction orthogonal to the local edge orientation, as shown in Figure 2. Define:

$$C_{fg}(p, q) = C_{fg}(q, p) = |f(x, y)| \cdot mPb(x, y) \tag{12}$$

$$\Theta_{fg}(p, q) = -\Theta_{fg}(q, p) = sign(f(x, y)) \cdot \phi \tag{13}$$

where ϕ represents a constant angular separation. These equations state that p and q should be embedded at angular separation ϕ with confidence that increases with figure/ground classifier confidence and edge strength. ϕ must be chosen sufficiently small such that the number of figure/ground layers in the image does not exceed $\frac{\pi}{\phi}$. We set $\phi = \frac{\pi}{8}$ in experiments.

By choosing r greater than the radius used for local intervening contour affinities, (C_{pb}, Θ_{pb}) and (C_{fg}, Θ_{fg}) have no overlapping nonzero entries. Writing

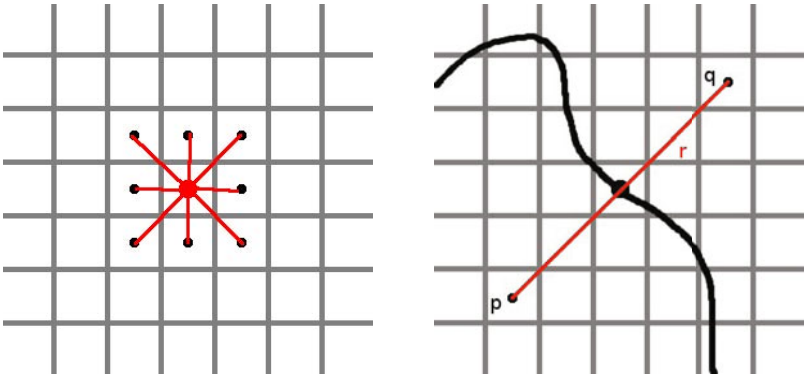


Fig. 2. Pairwise attraction and ordering. *Left:* We connect each pixel to its 8 immediate neighbors with affinity depending on the computed edge strength (pb) between them (the intervening contour [9]). A sparse matrix C_{pb} encodes these preferences. *Right:* The figure/ground classifier runs on a sampled set of nonmax-suppressed edge pixels. In each case, it induces a connection between the two pixels p and q located a fixed distance r from the edge point, in the direction orthogonal to the edge orientation. The predicted figural side defines a relative ordering $\Theta_{fg}(p,q)$, with an associated confidence $C_{fg}(p,q)$. Measurement matrix $M = C_{pb} + \alpha C_{fg} \bullet e^{i\Theta_{fg}}$ (up to a normalization factor) encodes both types of information, where $\bullet$ denotes element-wise product.

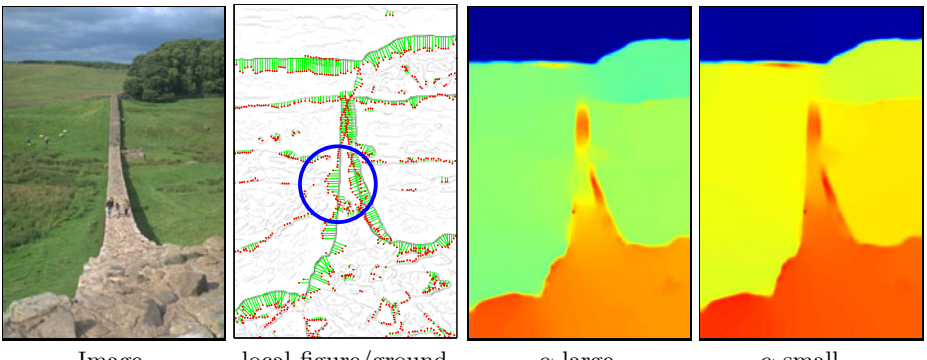


Image local figure/ground α large α small

Fig. 3. Competing segmentation and figure/ground cues. Where segmentation and local figure/ground predictions disagree (*blue circle*), the relative weighting, α, of the cues determines which dominates. For large α, locally incorrect figure/ground classification (*middle left*) overrules the tendency towards coherent ordering within regions, resulting in incorrect figure/ground globalization (*middle right*). For smaller α, adherence to strong segment boundaries corrects local figure/ground errors (*far right*). Setting $\alpha = 0$ results in recovery of segmentation only and ignores figure/ground ordering (*not shown*).

$C = C_{pb} + \alpha C_{fg}$ and $\Theta = \Theta_{pb} + \Theta_{fg}$, with α weighting the relative importance of the two signals, a measurement matrix M captures all information (excluding the normalization term involving C):

$$M = C \bullet e^{i\Theta} = C_{pb} \bullet e^{i\Theta_{pb}} + \alpha C_{fg} \bullet e^{i\Theta_{fg}} = C_{pb} + \alpha C_{fg} \bullet e^{i\Theta_{fg}} \tag{14}$$

The short-range connections (C_{pb}, Θ_{pb}) encode the prior that there is no figure/ground difference between neighboring pixels, but the confidence in this prior decreases in the vicinity of a strong edge. The longer-range connections (C_{fg}, Θ_{fg}) encode relative figure/ground ordering between more distant pixels. Attempting to satisfy both constraints yields an embedding which can violate the uniform prior near boundaries (where its confidence is low), in order to break apart figure and ground regions. Conversely, figure/ground differences are suppressed within a segment. Figure 3 demonstrates this type of competition.

2.4 Eigenvector Interpretation

Recall from Section 2.2 that the angle of the leading complex-valued eigenvector, $\angle z_0$, assigns each pixel a global figural ordering when solving the Angular Embedding problem specified by (14). We are left with the question of how to extract a segmentation. Note that in the absence of figure/ground cues, $C_{fg} = 0$ and $\Theta_{fg} = 0$, M is real-valued, and we find $\angle z_0 = 0$. Looking at the first $n+1$ eigenvectors $\{z_0, z_1, ..., z_n\}$ and their corresponding eigenvalues, $\lambda_0 \leq \lambda_1 \leq ... \leq \lambda_n$, we find a similar situation as for Normalized Cuts (2). In particular, $\lambda_0 = 0$, but the remaining eigenvectors provide an embedding in which segmentation is easy. For the general case with figure/ground cues, the now complex-valued eigenvectors still provide such an embedding (though with $\lambda_0 \neq 0$ and z_0 nontrivial).

We therefore extend the idea of extracting contours by computing gradients on the stack of eigenvector images [17] to the complex-valued case. As Figure 4 shows, we compute the "spectral" contour signal:

$$sPb(x, y, \theta) = \sum_{k=1}^{n} \frac{1}{\sqrt{\lambda_k}} \cdot \left([\nabla_\theta \Re\{z_k(x, y)\}]^2 + [\nabla_\theta \Im\{z_k(x, y)\}]^2 \right)^{\frac{1}{2}} \tag{15}$$

Following the procedure of Arbeláez *et al.* [3], we create a weighted combination of mPb and sPb and apply their Oriented Watershed Transform - Ultrametric Contour Map (OWT-UCM) algorithm to construct a hierarchical image segmentation. Averaging $\angle z_0$ over the resulting segments translates our figure/ground ordering on pixels into an ordering on regions.

3 Local Figure/Ground Classifier

The presentation so far has omitted the details of the figure/ground classifier introduced in Section 2.3. As previously mentioned, this classifier could predict depth ordering or a use a perceptual notion of figuralness. We choose the latter

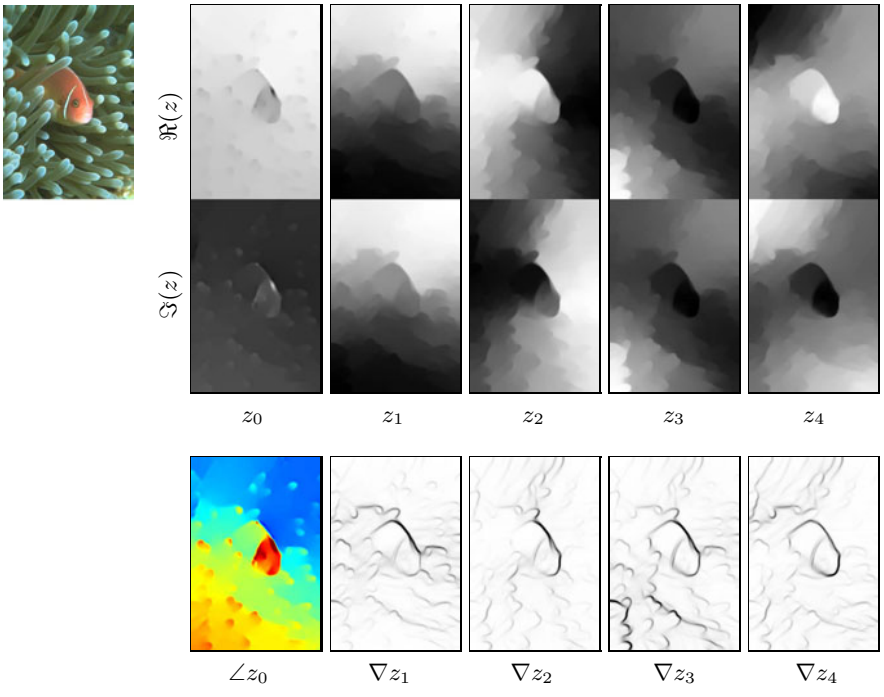


Fig. 4. Extracting figure/ground and segmentation from eigenvectors. *Top:* Real and imaginary components of the first five generalized eigenvectors, $\mathrm{z}_0, ..., \mathrm{z}_4$ obtained via Angular Embedding [31]. *Bottom Left:* Global figure/ground ordering is reported by $\angle \mathrm{z}_0$. *Bottom Right:* Maximum oriented gradients of eigenvectors, $\nabla z_k = \max_\theta\{\left([\nabla_\theta \Re\{z_k(x,y)\}]^2 + [\nabla_\theta \Im\{z_k(x,y)\}]^2\right)^{\frac{1}{2}}\}$, encode a global contour signal (shown here) from which we construct a segmentation.

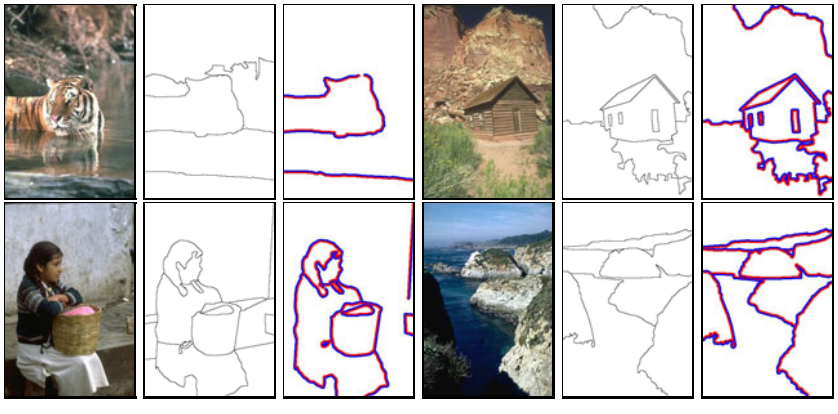


Fig. 5. Berkeley segmentation dataset (BSDS) with figure/ground labeling. *Left to Right:* Image, segment boundaries, and figure/ground annotations on a subset of those boundaries according to a human subject. Red marks the figural side. We use pre-existing segment [19] and figure/ground [10] labeling.

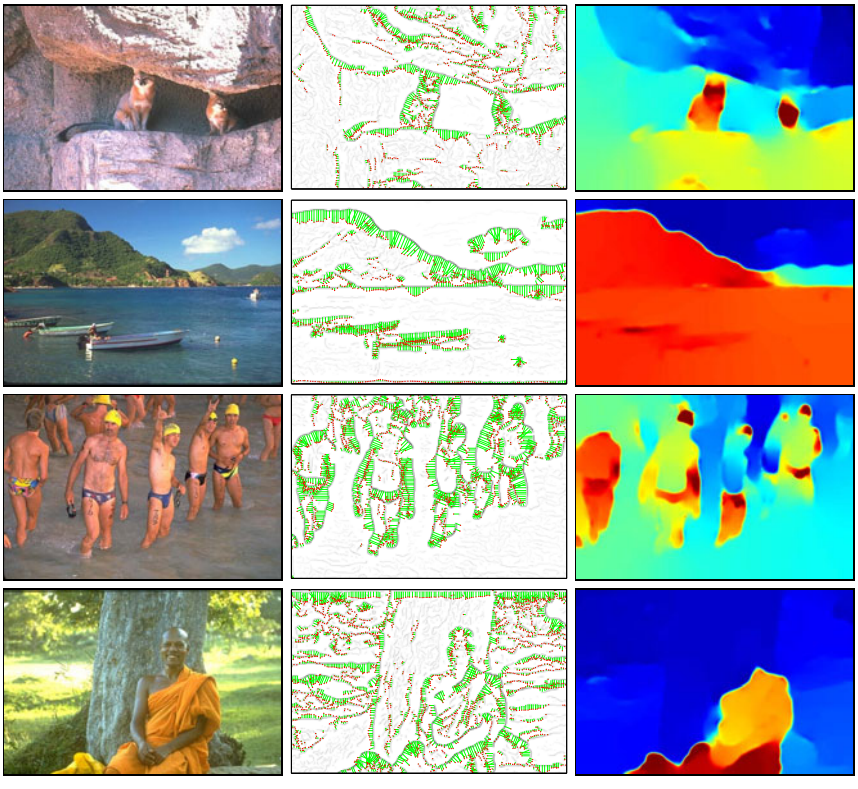


Fig. 6. Local to global figure/ground. *Left:* Image. *Middle:* Local figure/ground assignment by our shape-based classifier for the most salient mPb [17] contours. Vectors drawn from edge points indicate the predicted figural side by their red tip. Vector length corresponds to classifier confidence. *Right:* Recovered global figural ordering.

definition for convenience, keeping in mind that the primary focus of this paper is the new globalization algorithm for coupling figure/ground organization to segmentation, and not the engineering of this local classifier.

Rather than hand-design features for the figure/ground classifier, we borrow the approach of Ren *et al.* [24] and compute Geometric Blur [4] descriptors on top of the local mPb contour signal. We rotate each descriptor according to a local orientation estimate in an imperfect attempt to build-in limited rotation invariance (our final learned figure/ground classifier is not fully rotation invariant). Clustering these descriptors using K-means (with $K = 64$) yields a vocabulary of shapemes [24], which capture local contour configuration. A point of interest on a test contour is described by the vector measuring the similarity of its Geometric Blur descriptor to each of the shapemes.

We transfer human figure/ground labeling to the automatically generated nonmax-suppressed mPb contours using bipartite matching of edge pixels. We then train a logistic-regression classifier f that predicts local figure/ground assignment using the vector of shapeme similarities. This learned classifier performs at 62% accuracy, similar to the 65% accuracy reported by Ren *et al.* [24] for their local classifier. Figure 5 shows example human-annotated training data for this task and Figure 6 demonstrates local figure/ground predictions and recovered global ordering. In order to take only fairly reliable predictions into account during globalization, we sample edge locations (x, y) for which $mPb(x, y) > \tau$ and only run the local figure/ground classifier at those locations. We set $\tau = 0.3$.

4 Experiments and Discussion

Figures 7 and 8 show output of our algorithm for segmentation and figure/ground ordering on images from the Berkeley segmentation dataset (BSDS). Figure 9 compares our automatically generated segmentations to those of other algorithms [8,7,6,2,3] using the standard BSDS boundary precision-recall benchmark [18]. Precision-recall curves for the other algorithms are those reported in [3]. Our segmentations are better than all except those of the leading gPb-owt-ucm algorithm [3], to which they are fairly close. Though our algorithm can be seen a generalization of gPb-owt-ucm, there are a few technical differences in our implementation that may explain the small performance gap. One such difference is that we compute affinities only between neighboring pixels, reserving long-range connections for ordering cues, rather than use a larger radius for the intervening contour computation.

Not captured by these benchmarks is the fact that our system is the only one to solve for figure/ground. Moreover, our figure/ground output is not just a local determination of the figural side of each boundary, but is a global ranking of the regions in the segmentation. Our system offers the benefit of transforming a local figure/ground property defined on contours into a global one defined on regions. This may prove useful as a salience measure. For example, using only our bottom-up cues, the face automatically pops out as a figural region in the last example in Figure 7.

Our global figure/ground ordering comes at minimal additional computation cost over the segmentation-only approach. The local figure/ground classifier is fast to run on sampled edge points and computing eigenvectors for Angular Embedding is of the same complexity as computing them for Normalized Cuts.

Acknowledgments. Thanks to Stella X. Yu and Pietro Perona for helpful discussions.

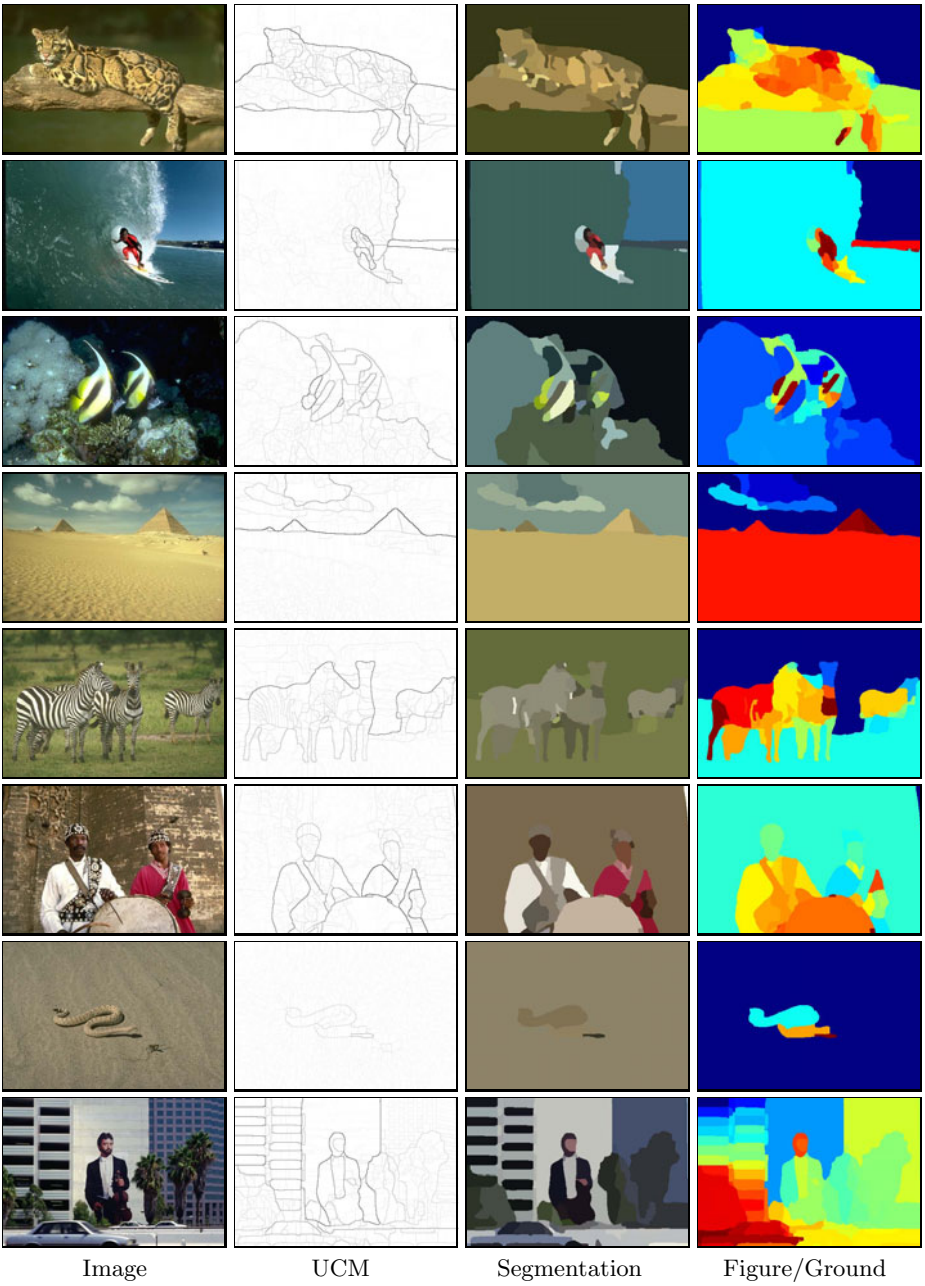


Fig. 7. Hierarchical segmentation and figure/ground ordering results. Our algorithm simultaneously generates a hierarchical image segmentation and a global figural ranking of regions. *Left:* Image. *Middle Left:* Hierarchical segmentation represented as an Ultrametric Contour Map (UCM) [2]. *Middle Right:* Regions at the optimal segmentation threshold displayed with their average color. *Right:* Global figure/ground ordering of the same regions. Red indicates more figural. All images shown belong to the test set.

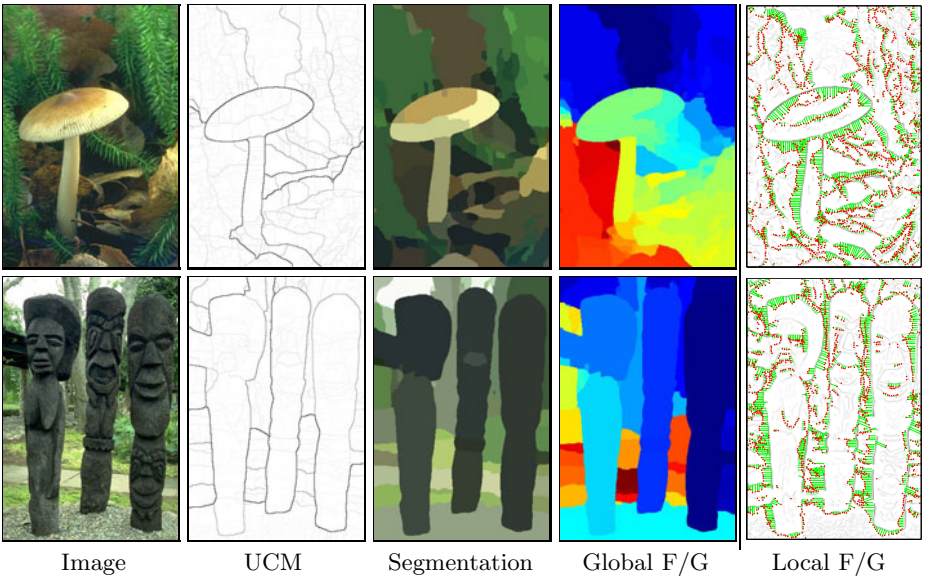


Fig. 8. Figure/ground failure examples. *Left to Right:* Image, UCM, segmentation, global and local figure/ground. Globalization errors occur when the local figure/ground classifier is consistently wrong over long contours (*e.g.* the left side of the mushroom or the sides of the statues). Note that good segment boundaries are still recovered.

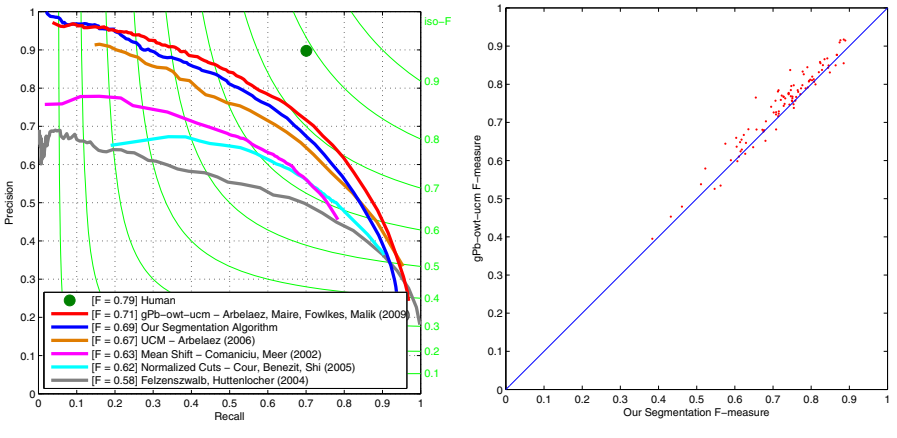


Fig. 9. Evaluation of region boundaries on the BSDS Benchmark. *Left:* The segmentation quality of our algorithm is close to that of the current best-performing algorithm, gPb-owt-ucm [3], and superior to others [8,7,6,2], as benchmarked by [3]. Algorithms are evaluated in terms of precision and recall with respect to human ground-truth boundaries. The maximum F-measure ($\frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$) is a summary score. Iso-F curves are shown in green. The dot indicates average human agreement. Our system is the only one that also solves for figure/ground. *Right:* Plotting per-image F-measures shows our segmentations to be competitive with those of gPb-owt-ucm.

References

1. Alpert, S., Galun, M., Basri, R., Brandt, A.: Image segmentation by probabilistic bottom-up aggregation and cue integration. In: CVPR (2007)
2. Arbeláez, P.: Boundary extraction in natural images using ultrametric contour maps. In: POCV (2006)
3. Arbeláez, P., Maire, M., Fowlkes, C., Malik, J.: From contours to regions: An empirical evaluation. In: CVPR (2009)
4. Berg, A.C., Malik, J.: Geometric blur for template matching. In: CVPR (2001)
5. Canny, J.: A computational approach to edge detection. PAMI (1986)
6. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. PAMI (2002)
7. Cour, T., Benezit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. In: CVPR (2005)
8. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. In: IJCV (2004)
9. Fowlkes, C., Martin, D., Malik, J.: Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches. In: CVPR (2003)
10. Fowlkes, C., Martin, D., Malik, J.: Local figure/ground cues are valid for natural images. Journal of Vision (2007)
11. Geiger, D., Kumaran, K., Parida, L.: Visual organization for figure/ground separation. In: CVPR (1996)
12. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV (2009)
13. Heitger, F., von der Heyd, R.: A computational model of neural contour processing: Figure-ground segregation and illusory contours. In: ICCV (1993)
14. Hoiem, D., Efros, A.A., Hebert, M.: Automatic photo pop-up. In: SIGGRAPH (2005)
15. Hoiem, D., Stein, A.N., Efros, A.A., Hebert, M.: Recovering occlusion boundaries from a single image. In: ICCV (2007)
16. Leichter, I., Lindenbaum, M.: Boundary ownership by lifting to 2.1D. In: ICCV (2009)
17. Maire, M., Arbeláez, P., Fowlkes, C., Malik, J.: Using contours to detect and localize junctions in natural images. In: CVPR (2008)
18. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color and texture cues. PAMI (2004)
19. 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)
20. Palmer, S.: Vision Science - From Photons to Phenomenology. MIT Press, Cambridge (1999)
21. Pao, H.K., Geiger, D., Rubin, N.: Measuring convexity for figure/ground separation. In: ICCV (1999)
22. Peterson, M.A., Gibson, B.S.: Must figure-ground organization precede object recognition? An assumption in peril. Psychological Science (1994)
23. Rao, S., Mobahi, H., Yang, A., Sastry, S., Ma, Y.: Natural image segmentation with adaptive texture and boundary encoding. In: Zha, H., Taniguchi, R.-i., Maybank, S. (eds.) ACCV 2009. LNCS, vol. 5994, pp. 135–146. Springer, Heidelberg (2010)
24. Ren, X., Fowlkes, C., Malik, J.: Figure/ground assignment in natural images. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 614–627. Springer, Heidelberg (2006)

25. Saund, E.: Perceptual organization of occluding contours of opaque surfaces. CVIU Special Issue on Perceptual Organization (1999)
26. Saxena, A., Chung, S.H., Ng, A.Y.: 3-d depth reconstruction from a single still image. IJCV (2008)
27. Sharon, E., Galun, M., Sharon, D., Basri, R., Brandt, A.: Hierarchy and adaptivity in segmenting visual scenes. Nature 442, 810–813 (2006)
28. Shi, J., Malik, J.: Normalized cuts and image segmentation. PAMI (2000)
29. Vecera, S.P., Vogel, E.K., Woodman, G.F.: Lower region: A new cue for figure-ground assignment. Journal of Experimental Psychology: General (2002)
30. Vese, L.A., Chan, T.F.: A multiphase level set framework for image segmentation using the mumford and shah model. IJCV (2002)
31. Yu, S.X.: Angular embedding: from jarring intensity differences to perceived luminance. In: CVPR (2009)
32. Yu, S.X., Lee, T.S., Kanade, T.: A hierarchical markov random field model for figure-ground segregation. In: CVPR (2001)
33. Yu, S.X., Shi, J.: Segmentation with pairwise attraction and repulsion. In: ICCV (2001)
34. Zhou, H., Friedman, H.S., von der Heydt, R.: Coding of border ownership in monkey visual cortex. Journal of Neuroscience (2000)

Cosegmentation Revisited: Models and Optimization

Sara Vicente[1], Vladimir Kolmogorov[1], and Carsten Rother[2]

[1] University College London
[2] Microsoft Research Cambridge

Abstract. The problem of cosegmentation consists of segmenting the same object (or objects of the same class) in two or more distinct images. Recently a number of different models have been proposed for this problem. However, no comparison of such models and corresponding optimization techniques has been done so far. We analyze three existing models: the L1 norm model of Rother et al. [1], the L2 norm model of Mukherjee et al. [2] and the "reward" model of Hochbaum and Singh [3]. We also study a new model, which is a straightforward extension of the Boykov-Jolly model for single image segmentation [4].

In terms of optimization, we use a Dual Decomposition (DD) technique in addition to optimization methods in [1,2]. Experiments show a significant improvement of DD over published methods. Our main conclusion, however, is that the new model is the best overall because it: (i) has fewest parameters; (ii) is most robust in practice, and (iii) can be optimized well with an efficient EM-style procedure.

1 Introduction

The task of Figure-Ground segmentation is a widely studied problem in computer vision. Given a *single* image there are techniques that attempt to automatically partition the image into multiple objects and background. If the goal is to have a single object segmented, i.e. a binary segmentation, there is the natural ambiguity of which object is the desired one. In this case interactive segmentation techniques must be considered where the user gives additional hints.

There are many interesting application scenarios where *multiple* images are available. This means each image depicts the "same" foreground object in front of potentially arbitrary backgrounds. In contrast to the single image case, the task of segmenting the common object automatically in all images is now well-defined. This task is called "cosegmentation" and was first addressed in [1]. Let us be more precise on the definition of the "same" foreground object. In this paper we use the definition of [1,2,3] where the only constraint is that the distribution of some appearance features of the foreground region in each image have to be similar. The appearance features can encode different information, like color and texture, and various similarity measures can be envisioned. This definition allows for a wide range of applications. One application is to create a visual summary from personal photo collections, by segmenting automatically all instances of the same object, e.g. a person and a dog [5]. Another application is to use the segmentation

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 465–479, 2010.

of the common object to efficiently edit all occurrences of this object in one step, e.g. by changing its contrast [6]. The practical challenge in the case of segmenting the same object is that distributions may not match exactly, due to changes in illumination, in viewpoint or object (self-)occlusion. Our definition of cosegmentation can potentially also be used for segmenting different objects of the same class. An example of an unsupervised object-class recognition and segmentation system is [7], where more features are used other than appearance, e.g. shape. It can be expected that for most object classes, appearance features alone are not strong enough, hence this application is out of the scope of this paper.

Very recently in [8] the authors used a different formulation of the cosegmentation problem. They casted it into a clustering problem with two cluster. They show results for image pairs and for multiple images of objects of the same class.

It is worth mentioning that several recent papers considered a simplified cosegmentation problem where user interaction is available. In [5] the authors segment several images of the same object, assuming one of those images is hand-segmented. They model local appearance and edge profiles from the segmented image in order to "transduct" such segmentation into the remaining images. In [9,6] the user input is in the form of foreground/background scribbles in one or many images from the collection. In [9] the authors discuss how the choice of the seed image influences the performance of their method. In [6] a way of guiding the user interactions is presented. We envision that the insights of this paper will also help to improve the task of interactive cosegmentation.

The goal of this paper is to examine theoretically and practically different models and optimization methods for cosegmentation. To achieve this we limit ourselves to the task of cosegmenting two images only, with color as the only appearance feature, and where distributions are expressed in terms of histograms. We consider three existing models [1,2,3], which differ only in the distance measure between the two color histograms. We also consider a new model, which is a straightforward extension from a single to multiple images of Boykov-Jolly [4]. For a fair comparison we improved on existing optimization methods for the models in [1,2]. We achieved this by using a *Dual Decomposition* technique. For a quantitative comparison we built a dataset of 100 image-pairs with varying levels of complexity by simulating changes in scale and illumination.

The paper is organized as follows. Section 2 introduces the four different models and discusses some of their properties. Since the optimization for some models is NP-hard, it is important to choose the best possible optimization procedure. In Sect. 3 we review such methods. In Sect. 4 we compare experimentally both the models and the optimization methods and conclude which are the better performing methods.

2 Models

We start this section by introducing some notation:

- $x_p \in \{0,1\}$ is the label for pixel p, where $p \in \mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$ and $\mathcal{P}_1$, $\mathcal{P}_2$ are respectively the set of pixels in image 1 and image 2. We use letter $k \in \{1,2\}$ for denoting the image number.

- z_p is the appearance of pixel p (e.g. color or texture) and such measurement is quantized into a finite number of bins. Variable b ranges over histogram bins ($b \in \{1, ..., B\}$ where B is the total number of bins), and $\mathcal{P}_{kb}$ denotes the set of pixels p in image k whose measurement z_p falls in bin b.
- h_k is the empirical un-normalized histogram of foreground pixels for image k: it is a vector of size B with components $h_{kb} = \sum_{p \in \mathcal{P}_{kb}} x_p$.

As stated earlier, one of the goals of this paper is to compare different cosegmentation models that have been previously proposed. Such models fit into a single framework, where the cosegmentation problem is formulated as an energy optimization, with an energy of the following form:

$$E(\boldsymbol{x}) = \sum_p w_p x_p + \sum_{(p,q)} w_{pq} |x_p - x_q| + \lambda E^{global}(h_1, h_2) \tag{1}$$

Jointly, the first two terms form the traditional MRF term for both images, where w_p is the unary weight for each pixel and w_{pq} is the pairwise weight. The last term, E^{global}, encodes a similarity measure between the foreground histograms of both images and λ is the weight for that term.

Following [1], we will use a ballooning term for the first term, constant for every pixel: $w_p = \mu$. This biases the solution to one of the possible labels and it is important to prevent trivial solutions (i.e. both images being labeled totally background or foreground). If the bias is not present (i.e. if $w_p = 0$ and the energy does not have unary terms) such trivial solutions are always a global optimum of the energy. Alternatively, in [2,3] the authors used user interaction to compute pixel-dependent unary terms [10]. We are interested in automatic cosegmentation so unary terms based on user interaction are not available.

The second term is a contrast sensitive smoothness term whose weight is given by $w_{pq} = \frac{\left(\lambda_i + \lambda_c \exp -\beta \|z_p - z_q\|^2\right)}{\mathrm{dist}(p,q)}$ with $\beta = \left(2 \left\langle (z_p - z_q)^2 \right\rangle\right)^{-1}$, where $\langle \cdot \rangle$ denotes expectation over the image and λ_i, λ_c are respectively the weight for Ising prior and for the contrast sensitive term.

The models differ in the way the term E^{global} in equation (1) is defined.

Model A: L1-norm. This model was first introduced in [1] and it was derived from a generative model. The global term in the energy was defined as follows:

$$E^{global} = \sum_b |h_{1b} - h_{2b}| \tag{2}$$

where the L1-norm is used to compute foreground histograms similarity.

Model B: L2-norm. This formulation was introduced in [2] and it was defined as follows:

$$E^{global} = \sum_b (h_{1b} - h_{2b})^2 \tag{3}$$

It is similar to the previous formulation in equation (2), with the difference that the norm used to measure histogram similarity is the L2-norm instead of the L1-norm. The authors motivate this change by arguing that such a model has some interesting properties and allows the use of alternative optimization methods.

Model C: Reward model. In [3] the authors used the following global term:

$$E^{global} = -\sum_b h_{1b} \cdot h_{2b} \tag{4}$$

They motivate the use of such a model by replacing the penalization term with a rewarding term.

Recall that the original formulation in [2,3] uses pixel-dependent unary terms, while we use a constant ballooning force: $w_p = \mu$.

Both model A and model B lead to NP-hard optimization problems [1], while model C leads to a submodular problem that can be efficiently optimized with graph cuts [3].

Model D: Boykov-Jolly model. The last model that we consider is a natural extension of the generative model for binary image segmentation in [4,1,11]. These papers use a separate appearance model for each of the two regions (background and foreground). In our case we have *three* regions - two separate backgrounds and one common foreground. Accordingly, we introduce three appearance models - θ_1^B, θ_2^B and θ^F. This leads to a generative model with the posterior described by the following energy function:

$$E\left(\boldsymbol{x}, \theta_1^B, \theta_2^B, \theta^F\right) = \sum_{(p,q)} w_{pq}|x_p - x_q| + \lambda \sum_k \sum_{p \in \mathcal{P}_k} U(x_p, \theta_k^B, \theta^F) \tag{5}$$

where

$$U(x_p, \theta^B, \theta^F) = \begin{cases} -\log(Pr(z_p|\theta^F)) & \text{if } \; x_p = 1 \\ -\log(Pr(z_p|\theta^B)) & \text{if } \; x_p = 0 \end{cases} \tag{6}$$

Since we are interested in automatic cosegmentation, the appearance models θ_1^B, θ_2^B and θ^F are not available in advance. In order to compute them, we minimize energy (5) jointly over segmentation and appearance models using an EM-style technique proposed in [11].

Model D is quite similar to the model used by Batra et al. [6] for interactive cosegmentation; the only difference is that Batra et al. used a single background model for all images. Model D also bears some resemblance to the generative model of Rother et al. [1] but there are some differences. In [1], the motivation was model selection, since two competitive models were considered: one where both images shared the same foreground appearance model and another where they had independent appearance models. The segmentation was then chosen so that the first model had higher posterior probability. In our case, we consider only a single model and try to find jointly the segmentation and appearance models that maximize the posterior probability. This formulation should be more appropriate when we know in advance that the two images have a common object. Also, it appears to lead to a simpler optimization problem: generalizing an EM-style procedure to the model in [1] is not straightforward.

2.1 An Alternative Formulation of Model D

To gain more insights into model D, we express its energy in a different way using the approach in [12]. It is known that for a fixed segmentation $\boldsymbol{x}$, optimal histograms that minimize energy (5) are simply the empirical histograms:

$$\theta_b^F = \frac{h_{1b} + h_{2b}}{H_1 + H_2} \qquad \theta_{kb}^B = \frac{\overline{h}_{kb}}{\overline{H}_k} \tag{7}$$

where we introduced the following notation: $H_k = \sum_b h_{kb}$ is the total number of foreground pixels in image k, $\overline{h}_{kb} = |\mathcal{P}_{kb}| - h_{kb}$ is the number of background pixels in image k belonging to bin b, and $\overline{H}_k = |\mathcal{P}_k| - H_k$ is the total number of background pixels in image k. Note, all quantities h_{kb}, $\overline{h}_{kb}$, H_k, $\overline{H}_k$ are functions of the segmentation $\boldsymbol{x}$ (recall that $h_{kb} = \sum_{p \in \mathcal{P}_{kb}} x_p$).

Following [12], we plug histograms (7) into the energy (5). Then the energy becomes of the form (1) with no unary terms ($w_p = \mu = 0$) and the following global term:

$$E^{global} = \sum_b \beta\left(h_{1b} + h_{2b}\right) + \sum_{k,b} \beta\left(\overline{h}_{kb}\right) - \beta\left(H_1 + H_2\right) - \sum_k \beta\left(\overline{H}_k\right) \tag{8}$$

where $\beta(z) = -z \log z$ is a concave function.

In the case of a single image the Boykov-Jolly model prefers assigning pixels in the same bin either entirely to the background or entirely to the foreground [12]; this leads to "compact" histograms. A similar fact holds for model D (the proof is entirely analogous to that in [12]).

Proposition 1. *Function (8) has a minimizer $\boldsymbol{x}$ such that for each (k, b), pixels in $\mathcal{P}_{kb}$ are either all labeled as 0 or all labeled as 1.*

2.2 Remarks on Model Properties

Before presenting an experimental comparison of the models, we would like to give some informal remarks which may give insights into their relative performance. We will first consider models A, B and D, and come back to model C at the end.

We believe that a fundamental difference of model D from other models is that it takes into account the prior knowledge that all regions are represented by compact histograms. For the case of a single image, the bias of the Boykov-Jolly model was discussed in [12]: it prefers segmentations in which pixels that fall in the same bin are assigned to the same segment (background or foreground), and among such segmentations the model picks the most *balanced* one, i.e. the segmentation in which the areas of the background and the foreground match. We conjecture that these properties carry over to the cosegmentation case. It can be shown, for example, that if the two images are identical and all bins are of the same size (i.e. $|\mathcal{P}_{kb}| = const$ for all k, b) then the global term will be minimized by a segmentation in which exactly half of the bins are assigned to the foreground. Due to a bias towards balanced segmentation we did not use the "ballooning force" for model D, i.e. we chose $\mu = 0$, which produced reasonable results. In contrast, the other models required this extra parameter μ in order to avoid trivial solutions.

Unlike model D, models A and B do not impose any penalty if pixels in the same bin, $\mathcal{P}_{kb}$, are assigned to two different segments. We argue that this has both pros and cons, as illustrated by two scenarios below.

Scenario 1. Assume that the background colors do not overlap with the foreground nor with the other background. Furthermore, suppose that the foreground regions in the two images match only partially, for example, due to an

illumination change or scaling. Thus, we have $|\mathcal{P}_{kb}| > |\mathcal{P}_{\overline{k}b}|$ for some bin b where $\overline{k} \in \{1,2\}, \overline{k} \neq k$. Models A and B will bias $|\mathcal{P}_{kb}| - |\mathcal{P}_{\overline{k}b}|$ pixels to an incorrect label. In contrast, model D should not be affected; it will assign all pixels in $\mathcal{P}_{kb}$ and $\mathcal{P}_{\overline{k}b}$ to the foreground, as desired.

Scenario 2. Let us now assume that we have "camouflage" in one of the images, i.e. colors of the background and the foreground overlap. Thus, we again have $|\mathcal{P}_{kb}| > |\mathcal{P}_{\overline{k}b}|$, but now the behavior of models A and B will be correct, while model D will try to incorrectly assign *all* pixels in $\mathcal{P}_{kb}$ to the foreground (or to the background).

We conclude that without camouflage model D should cope better with illumination and scale changes than model A and especially than model B. On the other hand, models A and B should be more robust to a camouflage in *one* of the images.

Let us now return to model C. Assume for simplicity that there are no pairwise terms. The energy can then be written as $E(\boldsymbol{x}) = \sum_b E_b(h_{1b}, h_{2b})$ where

$$E_b(h_{1b}, h_{2b}) = \mu(h_{1b} + h_{2b}) - \lambda h_{1b} \cdot h_{2b}$$

We must have $\mu > 0$, otherwise all pixels would be assigned to the foreground. Minimizing E_b over $[0, n_{1b}] \times [0, n_{2b}]$ where $n_{1b} = |\mathcal{P}_{1b}|$, $n_{2b} = |\mathcal{P}_{2b}|$ gives the following rule: if $n_{1b} \cdot n_{2b}/(n_{1b} + n_{2b}) \leq \mu/\lambda$ then assign pixels in $\mathcal{P}_{1b} \cup \mathcal{P}_{2b}$ to the background, otherwise assign these pixels to the foreground. This reliance on the harmonic mean of n_{1b} and n_{2b} can lead to unexpected results (Fig. 1). In our experiments we found that model C performs considerably worse than the other models.

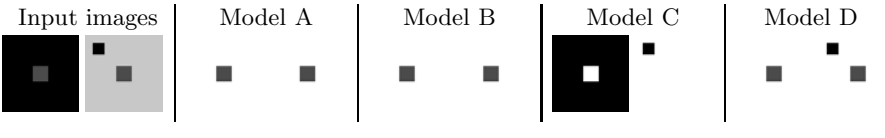


Fig. 1. Synthetic example illustrating the properties of the different models. The input images have only 3 different colors.

3 Optimization Methods

In this section we discuss several optimization methods that can be used for the models discussed in the previous section.

3.1 Trust Region Graph Cut (TRGC)

This method was proposed in [1] for model A and it can be viewed as a discrete analogue of trust region methods for continuous optimization. TRGC can be applied to energy functions of the form $E(\boldsymbol{x}) = E_1(\boldsymbol{x}) + E_2(\boldsymbol{x})$ where $E_1(\boldsymbol{x})$ is submodular and $E_2(\boldsymbol{x})$ is arbitrary. It works by iteratively replacing $E_2(\boldsymbol{x})$

with a linear approximation and it produces a sequence of solutions with the guarantee that in each iteration the energy does not go up.

In [1] the authors used TRGC inside an iterative scheme for cosegmentation that alternated between updating the segmentation for each image individually while the foreground histogram of the other image was fixed. This method requires a segmentation for initialization. In our experiments we observed that its performance is very dependent on that initialization.

We used the implementation of this method from [1]. We also adapted it to model B, i.e. replaced L1 norm with L2 norm.

3.2 Quadratic Pseudo Boolean Optimization

In [2] the authors observed that model B is represented by a **quadratic** pseudo-boolean function. Indeed, histograms h_1 and h_2 depend linearly on $\boldsymbol{x}$: $h_{kb} = \sum_{p \in \mathcal{P}_{kb}} x_p$. Therefore, expanding expression $(h_{1b} - h_{2b})^2$ yields a sum of linear terms and quadratic terms of the form $c_{pq} x_p x_q$, some of which are non-submodular. Mukherjee et al. [2] formulated a linear programming relaxation of the problem, which is equivalent to the roof duality relaxation [13,14] for the quadratic function $E(\boldsymbol{x})$. This relaxation can be solved via a maxflow algorithm, and it yields a partial solution: the nodes are divided into labeled and unlabeled, with the guarantee that the labels of the labeled nodes are optimal. An important question is how to set the segmentation for unlabeled nodes. Mukherjee et al. [2] use the segmentation obtained by minimizing energy $E(\boldsymbol{x})$ without the global term E^{global}. In our experiments we use a constant ballooning force ($w_p = \mu$), so this procedure assigns the same label to all unlabeled nodes.

Note that, model C is also represented by a quadratic function, but unlike the previous case this quadratic function is submodular. Therefore, model C can be optimized exactly by a single call to a maxflow algorithm [3].

3.3 Dual Decomposition (DD)

Dual Decomposition (DD) is a popular technique for solving combinatorial optimization problems [15], which proved to be very successful for MRF optimization [16,17,18,19,12]. The idea of DD is to decompose the original problem into smaller, easier subproblems that can be efficiently optimized. Combining the solution of such subproblems yields a lower bound for the initial problem. This lower bound is then maximized over different decompositions. We applied this technique to models A, B and D as described below.

Dual decompositions for models A and B. Let us write the corresponding optimization problems as follows:

$$\min_{\boldsymbol{x},\boldsymbol{y}} \quad E^{MRF}(\boldsymbol{x}) + \sum_b g(y_b) \tag{9a}$$

$$\text{s.t.} \quad y_b = \sum_{p \in \mathcal{P}_{1b}} x_p - \sum_{p \in \mathcal{P}_{2b}} x_p \equiv \sum_{k,p} a_{bp} x_p \qquad b = 1, ..., B \tag{9b}$$

where g is a convex function: $g(y) = \lambda|y|$ for model A and $g(y) = \lambda y^2$ for model B. Coefficients a_{bp} are defined as follows: $a_{bp} = 1$ if $p \in \mathcal{P}_{1b}$; $a_{bp} = -1$ if $p \in \mathcal{P}_{2b}$ and $a_{bp} = 0$ otherwise.

We form a standard Lagrangian function by relaxing constraints (9b) and introducing a Lagrangian multiplier $\boldsymbol{\theta}$:

$$L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = E^{MRF}(\boldsymbol{x}) + \sum_b g(y_b) + \sum_b \theta_b \left(y_b - \sum_p a_{bp} x_p \right) \tag{10}$$

Minimizing the Lagrangian over $(\boldsymbol{x}, \boldsymbol{y})$ gives a lower bound on the original problem:

$$\Phi(\boldsymbol{\theta}) = \min_{\boldsymbol{x},\boldsymbol{y}} L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) \tag{11a}$$

$$= \min_{\boldsymbol{x}} \left[E^{MRF}(\boldsymbol{x}) - \sum_{p,b} a_{bp} \theta_b x_p \right] + \sum_b \min_{y_b} \left[g(y_b) + \theta_b y_b \right] \tag{11b}$$

$$\Phi(\boldsymbol{\theta}) \leq E(\boldsymbol{x}) \tag{11c}$$

In order to obtain the tightest bound, we need to solve the following maximization problem:

$$\max_{\boldsymbol{\theta}} \Phi(\boldsymbol{\theta}) \tag{12}$$

This problem is dual to (9b). Function $\Phi(\boldsymbol{\theta})$ is concave; similar to [17,18,19], we use a subgradient method to maximize it. In order to compute a subgradient for a given vector $\boldsymbol{\theta}$, we need to solve $1 + B$ minimization subproblems in (11b). The first subproblem requires minimizing a submodular energy with pairwise terms, which can be efficiently done using graph cuts. Solving subproblems for bins b is straightforward.

It remains to specify how to choose a primal solution $\boldsymbol{x}$. Let $\boldsymbol{x}^t$ be a minimizer of the first subproblem in (11b) at step t of the subgradient method. Among labelings $\boldsymbol{x}^t$, we choose the solution with the minimum cost $E(\boldsymbol{x}^t)$.

Dual decompositions for model D. We obtained a lower bound by relaxing constraints $H_k = \sum_p x_k$ and using the fact that $\overline{H}_k \equiv |\mathcal{P}_k| - H_k$. Details are very similar to those in [12].

4 Experimental Results

In this section we describe the experimental results. We start by giving details on the setup used to compare the different models. In section 4.1, we compare the performance of the different optimization methods, and in section 4.2, using the best optimization procedure for each model, we compare the performance and robustness of such models.

Dataset. Given the difficulty in acquiring ground truth data for the cosegmentation problem, we used composites of 40 different backgrounds with 20 foreground objects from the database in [20], for which high quality alpha mattes are available. The database in [20] has more than 20 images; we selected

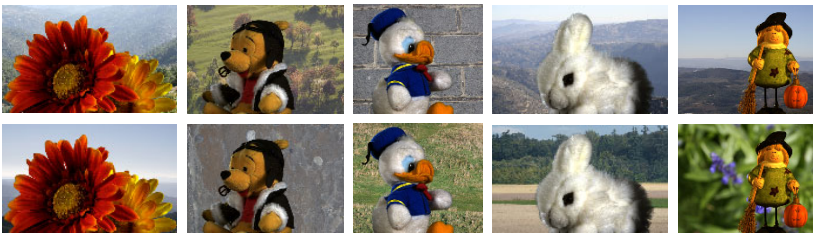


Fig. 2. Some of the images in the dataset. These images are composites using the same foreground.

objects with fewer transparencies. Representative images out of these 20 pairs are shown in Fig. 2.

We resized the images so that their maximum side is 150 pixels. Some of the models and optimization methods discussed are limited to small images, in particular, model C and QPBO. Both these optimization methods require the construction of graphs that grow quadratically with the size of the image.

The use of exactly the same foreground object in both images ensures that the histograms over pure foreground pixels match. The choice of such simplified dataset is justified by the intuition that if the models and optimization methods fail in this scenario, they will also fail in a realistic scenario where the foreground histograms may differ. In section 4.2 we also test more realistic scenarios by varying the size and illumination in one of the images.

Choosing weights μ and λ. The choice of weights for the different terms in the energy greatly affects the performance of the methods. We test the different models with different combinations of these weights. In order to reduce the search space, we fix $\lambda_i = 1$ and $\lambda_c = 50$ for all methods, similar to what is done in [1]. As for parameters λ and μ, we used leave-one-out cross-validation for each model, where parameters are allowed to take values in a discrete domain[1]. Results are given in section 4.2. For comparison, we also report results when the weights for each image are chosen optimally according to GT.

In section 4.1 we are only interested in comparing optimization methods, so we fix the weights in an ad-hoc way. For model A, we choose $\lambda = 5$ and $\mu = -2$, for model B, $\lambda = 2$ and $\mu = -10$ and for model D, $\lambda = 1$.

Histograms. We use histograms over RGB colors, using 16 bins for each color channel. Note that, in previous papers where some of the models were introduced,

[1] For model A and model B we test 16 different configurations, where $\lambda \in \{0.01, 0.1, 1, 10\}$ and $\mu \in \{-0.01, -0.1, -1, -10\}$. Since some of these configurations lead to trivial solutions, we handpick 8 other intermediate configurations that look more promising. Thus, there are 24 possible combinations of weights.

Model C allows the use of parametric maxflow for parameter learning. Fixing λ, we efficiently compute solutions for all possible values of μ using parametric maxflow. We test 4 different values for λ: 0.001, 0.01, 0.1 and 1.

Model D only has one free parameter, λ, and we test 12 different values for this weight: 0.01, 0.1, 0.5, 1, 2, 5, 10, 15, 20, 30, 40 and 100.

other appearance features were used [1,3]. Since our dataset is constructed such that the foreground histograms over color are very similar, we expect that none of the models is negatively affected by this choice of histogram quantization.

4.1 Results Comparison for Optimization Algorithms

Here we compare the optimization methods reviewed in section 3. We start by comparing Dual Decomposition with TRGC for models A and B. Since TRGC is an iterative method that requires as input an initial segmentation, we test this method with three different starting points. First, we use the solution of DD as a starting point. The second starting point is a random segmentation whose foreground histogram is constructed by having each bin take the minimum value over the corresponding bins in the full histogram of both images, i.e., $h_b = \min(|\mathcal{P}_{1b}|, |\mathcal{P}_{2b}|)$. Third, we initialize TRGC with the ground truth (GT). GT is not available at test time, and we report results only for comparison.

The results for model A are shown in the first part of Table 1. Note that in [1], where TRGC was proposed, DD was not used as a starting point. For this model, the difference between TRGC-DD and DD is very small, since TRGC starting with DD only improves the energy for two images.

Table 1. Comparison of optimization methods for Models A and B. We compare TRGC (using 3 different initial solutions), Dual Decomposition, and QPBO (only for model B). For each model, the first row shows for how many images each method gives the best energy. The second row is the gap between the energy and the lower bound (LB) obtained by DD. The values are normalized: first we add a constant to each term of the energy so that the minimum of each term becomes 0, and then scale the energy so that the lower bound corresponds to 100. The last row is the error rate: percentage of misclassified pixels over the total number of pixels.

		TRGC			DD	QPBO
		From DD	From hist	From GT		
Model A	Best energy: # cases	20	0	0	18	-
	Distance from LB	100.24	106.5	101.15	100.24	-
	Error rate	3.7%	8.1%	3.2%	3.7%	-
Model B	Best energy: # cases	13	0	7	3	0
	Distance from LB	101.59	107.56	101.77	104.20	197.29
	Error rate	3.93%	5.96%	2.85%	3.92%	51.77%

The results for model B are shown in the second part of Table 1. Although QPBO also provides a lower bound, we used the lower bound obtained by Dual Decomposition since in our experiments, it was always better than the one provided by QPBO.

We conclude that a combination of DD and TRGC, using DD solution as a starting point for TRGC, is the best performing method for both model A and B, and this is the method used in the next section for model comparison.

Surprisingly, the performance obtained for the QPBO method contrasts with the one reported in [2], since for this experiment the number of pixels left unlabeled by this method was 90%. Note that in [2], the authors used a different spatially varying unary term which may induce differences. They also report that the performance of the method deteriorates when weight λ is increased. In the case considered, where w_p is constant, small values of λ lead to trivial solutions.

In order to better understand why QPBO fails, we ran the method with a fixed ballooning force, $\mu = -10$, and different values of λ. In Table 2, we show the percentage of pixels that were labeled one, zero, or left unlabeled. For intermediate values of λ, the number of unlabeled pixels is more than 90%. For such values, QPBO is not reliable as an optimization method. On the other hand, for extreme values of λ, QPBO labels more pixels, but the resulting model is not meaningful, for example, for the case $\lambda = 10^{-3}$, all pixels for all images considered were labeled 1.

Table 2. QPBO results. Percentage of pixels labeled 1, 0 or left unlabeled by the QPBO method for different values of weight λ.

λ	10^{-3}	10^{-2}	10^{-1}	10^{0}	10^{1}	10^{2}	10^{3}
Labeled 1	100	64.49	9.52	0.18	0.03	0.03	0.03
Labeled 0	0	0	0	0	22.68	25.66	24.22
Unlabeled	0	35.51	90.48	99.82	77.30	74.31	75.75

Dual Decomposition for model D. We compared two different optimization methods for model D: the EM-style iterative procedure of [11] and a DD approach. For the EM-style optimization, we initialized the color models in the same way as discussed before for TRGC initialization when taking the histograms' intersection. Since DD provides a lower bound, we compared the gap between the lower bound and the energy obtained by both models. For DD this gap is 109.5 and for the EM-style optimization it is 103.4. The average gap is reduced to 103.2, if the best method is chosen for each image. This is very similar to the gap obtained by the iterative technique and we conclude that the improvement of using DD is only marginal for this problem and we report results using the EM-style optimization.

4.2 Results Comparison for Models

In this section we compare the four different models. We present results for three different cases. In the first case, we use the original images (some of the images are shown in Fig. 2), where the same foreground is composed with two different backgrounds. This is the simplest case and the error rate is reported in the first row of Table 3.

In the second case, we consider images of different sizes, reducing one of the images to 90% and 80% of the original size. This leads to a more complicated cosegmentation problem, where the object has different sizes in both images.

In the third case, in order to simulate illumination changes, we add a constant to all RGB values (ranging from 0 to 255) of one of the images. We show results for two different values of this constant: 3 and 6.

In Table 3, we also present the histogram similarity for the different cases. This similarity is given by: $100 - 100 \times \frac{\sum_b |h_{1b}^{GT} - h_{2b}^{GT}|}{\sum_b h_{1b}^{GT} + h_{2b}^{GT}}$ where h_k^{GT} is the histogram of image k computed over foreground ground truth pixels. This similarity can be seen as a rough measure of the difficulty of the problem, and the higher it is, the simpler the problem.

Table 3. Error rate using leave-one-out cross-validation. We compare the error rate for the different methods in 3 different scenarios. We also report the standard error of estimating the mean of the error rate. For the first case we use the original composites. In the second case we consider images of different sizes, reducing one of the images to 90% and 80% of the original size. In the third case, in order to simulate illumination changes, we add a constant to all RGB values of one of the images, 3 and 6. The last column shows the similarity of the foreground histograms of both images.

	Model A	Model B	Model C	Model D	Histogram similarity
Original images	4.6% ±0.8	3.9% ±0.7	22.0% ±3.9	4.3% ±0.3	93.4
Resized to 90%	4.7% ±0.4	5.7% ±0.8	16.3% ±2.4	4.9% ±0.5	84.6
Resized to 80%	7.8% ±1.3	9.7% ±1.4	17.4% ±3.0	5.1% ±1.0	74.2
RGB +3	4.4% ±0.4	7.1% ±1.1	21.4% ±4.3	3.7% ±0.3	84.6
RGB +6	5.5% ±0.5	12.3% ±1.7	20.3% ±2.5	4.0% ±0.4	76.3

From the results presented in Table 3 we take the following statistically significant observations:

- Models A, B, and D perform similarly for the simplest case.
- Model C is the worst performing model since it produces in every case considerably higher error rates.
- Model D is the most robust to changes in size and illumination.
- Comparing both models based on histogram distances, the L1-norm (Model A) is more robust than the L2-norm (Model B), for the cases where there are small variations of foreground.

Some methods may be affected negatively by the way the weighting parameters are chosen, since image measurements are not taken into account. In order to fairly compare the methods without introducing this type of bias, we also present results in Table 4 for the case where the weights λ and μ are chosen independently for each image, so that the error rate is minimized.

Table 4. Error rate without cross validation. These results correspond to choosing the best weights λ and μ according to GT for each image individually. They should be compared with Table 3.

	Model A	Model B	Model C	Model D	Histogram similarity
Original images	3.2% ±0.3	2.9% ±0.3	8.8% ±1.9	3.2% ±0.3	93.4
Resized to 90%	4.2% ±0.4	4.0% ±0.4	8.1% ±1.7	3.2% ±0.3	84.6
Resized to 80%	5.2% ±0.6	6.2% ±0.6	7.0% ±1.4	3.2% ±0.3	74.2
RGB +3	3.3% ±0.3	4.0% ±0.2	9.3% ±1.8	3.2% ±0.2	84.6
RGB +6	4.3% ±0.4	8.0% ±1.2	9.2% ±1.8	3.3% ±0.2	76.3

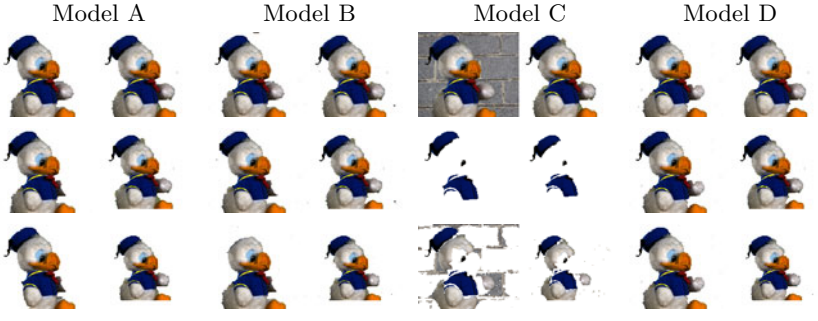


Fig. 3. Results without cross-validation. Segmentation obtained for each model when reducing the size of the second image.

Comparing tables 3 and 4, it can be seen that model C has the greatest improvement in error rate when the choice of weights is done independently for each image. However, it still remains the worst performing model.

In Fig. 3, we show some cosegmentation results for a pair of images for different sizes of the second image. The results shown agree with the insights discussed in Sect. 2.2. When the size of the images differ, both models A and B incorrectly cut some parts of the object, in order to improve the matching of the resulting foreground histograms. Model C gives unpredictable results due to the mentioned bias. Model D copes better with the changes in image size.

4.3 Results for Real Images

Following a reviewer's suggestion, we tested the different models on the real images used in [3][2].

[2] Images and GT are available from http://www.cs.wisc.edu/~vsingh/pairimages.tar.gz. We chose 20 pairs of images from this dataset, excluding the ones which were created in a similar way to our dataset.

We observed that the histogram quantization used in the rest of the paper is not appropriate for these images, since there are significant differences in the foreground color histograms and the overlap of the background color histograms is large. The overlap for the foreground histograms is 39% which is considerably lower than the overlap reported in the last column of Table 3. On the other hand, the overlap of background histograms is 21% compared to 8% for our dataset of composed images. This affected the results negatively and the error rates are between 20% and 30% for all image pairs. The use of better histogram quantization would considerably improve the performance for all methods.

This observation further supports our use of composed images, since the goal of the paper is to compare the performance of the different methods in a scenario where external factors with a negative impact could be easily controlled.

Note that, the results reported for the same images in [2,3] used user interaction and the results in [1] used various features to calculate the histograms.

5 Conclusions and Future Work

Recently, several models for cosegmentation have been proposed some of which lead to challenging optimization problems. We showed that they are outperformed by a natural extension of the Boykov-Jolly model, which has not been considered in the context of cosegmentation before. The improvement of model D over models B and especially C is substantial. The gap between models D and A is less significant, and potentially could be affected by the choice of a dataset. However, model D has two clear advantages: it has one less parameter, and it allows the use of an effective and fast EM-style optimization.

To enable a fair comparison of models, we had to improve on optimization techniques in [1,2]. We believe the Dual Decomposition method that we used for models A and B was adequate for our task. Although, we did not get verifiable global minima, the gap between the lower bound and the energy was small enough, and furthermore, using ground truth to initialize an iterative technique (TRGC) led to higher energies compared to DD.

In the future, we plan to gather a larger and more challenging dataset of images for cosegmentation, including the ones used in Sect. 4.3. The focus will be on the construction of discriminative histograms, that take into account not only color but also other features like SIFT and Gabor filters as in [8].

Acknowledgements. We thank Vikas Singh for answering questions about his implementation.

References

1. Rother, C., Kolmogorov, V., Minka, T., Blake, A.: Cosegmentation of image pairs by histogram matching - incorporating a global constraint into MRFs. In: CVPR (2006)
2. Mukherjee, L., Singh, V., Dyer, C.R.: Half-integrality based algorithms for cosegmentation of images. In: CVPR (2009)

3. Hochbaum, D.S., Singh, V.: An efficient algorithm for co-segmentation. In: ICCV (2009)
4. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In: ICCV (2001)
5. Cui, J., Yang, Q., Wen, F., Wu, Q., Zhang, C., Cool, L.V., Tang, X.: Transductive object cutout. In: CVPR (2008)
6. Batra, D., Kowdle, A., Parikh, D., Luo, J., Chen, T.: iCoseg: Interactive co-segmentation with intelligent scribble guidance. In: CVPR (2010)
7. Winn, J., Jojic, N.: Locus: learning object classes with unsupervised segmentation. In: ICCV (2005)
8. Joulin, A., Bach, F., Ponce, J.: Discriminative clustering for image co-segmentation. In: CVPR (2010)
9. Batra, D., Parikh, D., Kowdle, A., Chen, T., Luo, J.: Seed image selection in interactive cosegmentation. In: ICIP (2009)
10. Personal communication with Vikas Singh
11. Rother, C., Kolmogorov, V., Blake, A.: Grabcut - interactive foreground extraction using iterated graph cuts. In: SIGGRAPH (2004)
12. Vicente, S., Kolmogorov, V., Rother, C.: Joint optimization of segmentation and appearance models. In: ICCV (2009)
13. Hammer, P.L., Hansen, P., Simeone, B.: Roof duality, complementation and persistency in quadratic 0-1 optimization. Math. Programming 28, 121–155 (1984)
14. Boros, E., Hammer, P.L.: Pseudo-boolean optimization. Discrete Applied Mathematics 123(1-3), 155–225 (2002)
15. Bertsekas, D.: Nonlinear Programming. Athena Scientific, Belmont(1999)
16. Wainwright, M., Jaakkola, T., Willsky, A.: MAP estimation via agreement on trees: Message-passing and linear-programming approaches. IEEE Trans. Information Theory 51(11), 3697–3717 (2005)
17. Schlesinger, M.I., Giginyak, V.V.: Solution to structural recognition (MAX,+)-problems by their equivalent transformations. Part 1. In: Control Systems and Computers, pp. 3–15 (2007)
18. Schlesinger, M.I., Giginyak, V.V.: Solution to structural recognition (MAX,+)-problems by their equivalent transformations. Part 2. In: Control Systems and Computers, pp. 3–18 (2007)
19. Komodakis, N., Paragios, N., Tziritas, G.: MRF optimization via dual decomposition: Message-passing revisited. In: ICCV (2005)
20. Rhemann, C., Rother, C., Wang, J., Gelautz, M., Kohli, P., Rott, P.: A perceptually motivated online benchmark for image matting. In: CVPR (2009)

Optimal Contour Closure by Superpixel Grouping

Alex Levinshtein[1], Cristian Sminchisescu[2], and Sven Dickinson[1]

[1] University of Toronto
{babalex,sven}@cs.toronto.edu
[2] University of Bonn
cristian.sminchisescu@ins.uni-bonn.de

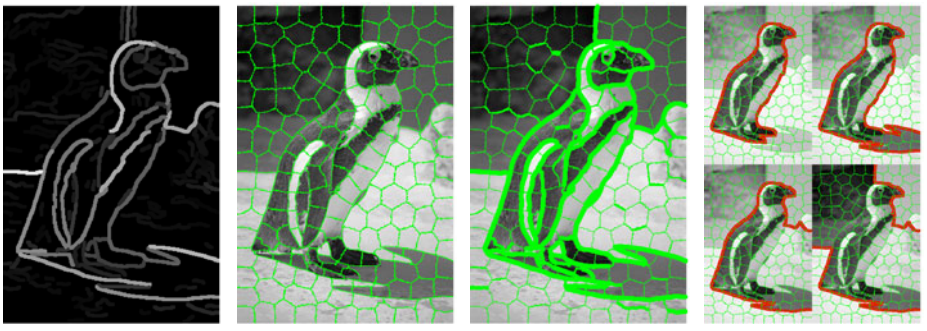
Abstract. Detecting contour closure, i.e., finding a cycle of disconnected contour fragments that separates an object from its background, is an important problem in perceptual grouping. Searching the entire space of possible groupings is intractable, and previous approaches have adopted powerful perceptual grouping heuristics, such as proximity and co-curvilinearity, to manage the search. We introduce a new formulation of the problem, by transforming the problem of finding cycles of contour fragments to finding subsets of superpixels whose collective boundary has strong edge support in the image. Our cost function, a ratio of a novel learned boundary gap measure to area, promotes spatially coherent sets of superpixels. Moreover, its properties support a global optimization procedure using parametric maxflow. We evaluate our framework by comparing it to two leading contour closure approaches, and find that it yields improved performance.

1 Introduction

One of the key challenges in perceptual grouping is computing contour closure, i.e., linking together a set of fragmented contours into a cycle that separates an object from its background. What makes the problem particularly hard is the intractable number of cycles that may exist in the contours extracted from an image of a real scene. Early perceptual grouping researchers [1] identified a set of nonaccidental contour relations, such as symmetry, parallelism, collinearity, co-curvilinearity, etc., that can be used to link together causally related contours. Such nonaccidental grouping rules can serve as powerful heuristics to help manage the complexity of greedily searching for a contour closure that is unlikely to have arisen by chance [2,3]. However, the space of possible closures is still overwhelming, particularly when one allows larger and larger boundary gaps in a closure. Finding an optimal solution is intractable without somehow reducing the complexity of the problem.

In this paper, we introduce a novel framework for efficiently searching for an optimal closure. Fig. 1 illustrates an overview of our aproach. Given an image of extracted contours (Fig. 1(a)), we begin by restricting contour closures to pass along boundaries of superpixels computed over the contour image (Fig. 1(b)).

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 480–493, 2010.



(a) (b) (c) (d)

Fig. 1. Approach Overview: (a) contour image – while we take only contours as input, we will overlay the original image in the subsequent figures for clarity; (b) superpixel segmentation, in which superpixel resolution is chosen to ensure that target boundaries are reasonably well approximated by superpixel boundaries; (c) a novel, learned measure of gap reflects the extent to which the superpixel boundary is supported by evidence of a real image contour (line thickness is inversely proportional to gap); (d) our cost function can be globally optimized to yield the largest set of superpixels bounded by contours that have the least gaps. In this case the solutions, in increasing cost (decreasing quality), are organized left to right, top to bottom.

In this way, our first contribution is to reformulate the problem of searching for cycles of contours as the problem of searching for a subset of superpixels whose border has strong contour support in the contour image; the assumption we make here is that those salient contours that define the boundary of the object (our target closure) will align well with superpixel boundaries. However, while a cycle of contours represents a single contour closure, our reformulation needs a mechanism to prefer superpixel subsets that are spatially coherent.

Spatial coherence is an inherent property of a cost function that computes the ratio of perimeter to area. We modify the ratio cost function of Stahl and Wang [4] to operate on superpixels rather than contours, and extend it to yield a cost function that: 1) promotes spatially coherent selections of superpixels; 2) favors larger closures over smaller ones; and 3) introduces a novel, learned gap function measuring the agreement between the boundary of the selection and image contours. The third property adds cost as the number and sizes of gaps between contours increase. Given a superpixel boundary fragment (e.g., a side of a superpixel) representing a hypothesized closure component, we assign a gap cost that's a function of the proximity of nearby image contours, their strength, their orientation, and their curvature (Fig. 1(c)). It is in this third property that our superpixel reformulation plays a second important role – by providing an appropriate scope of contour over which our gap analysis can be conducted.

In our third and final contribution, the two components of our cost function, i.e., area and gap, are combined in a simple ratio that can be efficiently optimized

using parametric maxflow [5] to yield the global optimum. The optimal solution yields the largest set of superpixels bounded by contours that have the least gaps (Fig. 1(d)). Moreover, parametric maxflow will yield the top k solutions (see [6], for example). In an object recognition setting, generating a small set of such solutions can be thought of as generating a small set of promising shape hypotheses which, through an indexing process, could invoke candidate models that could be verified (detected).

In the following sections, we begin by reviewing related work on contour closure (Sec. 2). Next, in Sec. 3, we introduce our problem formulation that transforms the problem of finding optimal cycles of contour fragments into the problem of finding an optimal subset of superpixels. It is here that our cost function is described. In Sec. 4, we describe our process for learning our gap function from training data, and in Sec. 5, we present an efficient procedure for finding the global minimum of our cost function using parametric maxflow. In Sec. 6, we evaluate our framework, comparing it to two competing approaches for computing closure, and discuss the strengths and weaknesses of our approach. We also illustrate the third important role of our superpixel reformulation of providing an appropriate scope over which appearance can be analyzed, and show how our grouping framework can easily be augmented to include both contour and region information. Finally, in Sec. 7, we draw conclusions and outline our plans for future work.

2 Related Work

Detecting closed contours in an image has been addressed by many researchers in different ways. One possible taxonomy for categorizing related work is based on the nature of the prior information used to constrain the grouping process. We will stop short of reviewing methods which assume object-level priors, for it is unclear how to make such methods scale up to very large databases. Instead, we focus on methods that make no assumptions about scene content, although as we will see, many make assumptions about the nature of parts that make up the objects in the scene. In fact, some methods incorporate low-, mid-, and high-level shape priors, as exemplified by Ren et al. [7]. We will also stop short of reviewing methods focused solely on contour completion, e.g., Ren et al. [8] and Williams and Jacobs [9], although the regularities exploited by such approaches can clearly play a powerful role in detecting closure.

Many researchers have exploited the classical Gestalt cues of parallelism and symmetry to group contours. Lowe's [10] early work on perceptual grouping was one of the first to develop a computational model for parallelism, collinearity, and proximity. Many computational models exist for symmetry-based grouping, including Brady and Asada [11], Cham and Cipolla [12], Saint-Marc et al. [13], Ylä-Jääski and Ade [14], and more recently, Stahl and Wang [15]. One significant challenge faced by these systems is the complexity of pairwise contour grouping to detect symmetry-related contour pairs. Levinshtein et al. [16] attempt to overcome this computational complexity limitation by constraining the

symmetric parts to be collections of superpixels. We will draw on this idea of grouping superpixels, but will focus on the more generic cue of closure.

Further down the spectrum of prior knowledge are methods based on weaker shape priors than parallelism and symmetry. For example, Jacobs [17] uses convexity as well as gap to extract closed contours by grouping straight line segments. A less restrictive measure is that of compactness, which can be attained by normalizing the gap by area (Estrada and Jepson [2,3], Stahl and Wang [4]). Some measure of internal homogeneity can also be used (Estrada and Jepson [3], Stahl and Wang [4]), provided that the inside of the region is easily accessible.

Finally we come to the most general methods that compute closure using only weak shape priors, such as continuity and proximity. The most basic closure-based cost function uses a notion of boundary gap, which is a measure of missing image edges along the closed contour. Elder and Zucker [18] model the probability of a connection between two adjacent contour fragments, and find contour cycles using a shortest path algorithm. Wang et al. [19] optimize a measure of average gap using the ratio cut approach. However, a measure based purely on the total boundary gap is insufficient for perceptual closure, and Elder and Zucker [20] argue that the distribution of gaps along the contour is also very important. Williams and Hanson [21] addressed the problem of perceptual completion of occluded surfaces, formulated as the problem of computing a labeled knot-diagram representing a set of occluded surfaces from observed image contours. While formulated as an elegant combinatorial optimization problem, for which an optimal solution was available, the approach was not tested on real scenes.

All the above methods suffer from the high complexity of choosing the right closure from a sea of contour fragments. To cope with this complexity, they either resort to heuristics to prune the search (e.g., [17]) or constrain the search space by other means (e.g., restricting the closure to alternating gap/non-gap cycles [4]). Zhu et al. [22] propose to solve this hard grouping problem by embedding the edge fragments into polar coordinates such that closed contours correspond to circles in that space; however, their goal is to better detect object contours, and they stop short of grouping the contours into closed boundaries. The method of Jermyn and Ishikawa [23] is perhaps the closest to our work. Similar to [19,4], they minimize closure costs using ratio cuts, but unlike [19,4] who operate on contour fragments, [23] works directly with pixels in a 4-connected image grid. It enables the authors to minimize many different closure costs (including our own) by globally minimizing ratio cuts in a simply connected planar graph. However, individual pixels provide poor scope for gap computation. In contrast, our superpixels not only provide greater scope for gap computation (which, in our case, is learned), they provide greater scope for the incorporation of internal appearance-based affinity. Moreover, while their solution is optimal, it does not provide a set of optimal solutions that capture closures at multiple scales.

In this paper, our goal is to find closed contour groups in an efficient manner. Drawing on [16], we use superpixels to constrain the search space of the resulting closures. Superpixels also provide an easy way to access internal region information (such as region area). Moreover, superpixel boundaries provide better scope

for gap computation, as opposed to most previous methods that linearize the output of an edge detector or operate directly on image pixels. On the optimization side, we show that parametric maxflow [5] can be used not only to recover the global optimum of closure costs similar to that of Stahl and Wang [4] and Jermyn and Ishikawa [23], but can also be used to recover a multiscale set of closure hypotheses.

3 Problem Formulation

As mentioned in Sec. 1, our framework reduces grouping complexity by restricting closure to lie along superpixel boundaries. Given a contour image $I(x, y)$[1], we first segment it into N superpixels using a modified version of the superpixel segmentation method of Mori et al. [25] ([25] uses the Pb edge detector [26], while we use globalPb [24]). If we let X_i be a binary indicator variable for the i-th superpixel, the vector $\boldsymbol{X}$ yields a full labeling of the superpixels of I as figure (1) or ground (0). Recall that our goal will be to select a maximal set of superpixels which have high spatial coherence and whose boundary has strong contour support in the image. Drawing on Stahl and Wang [4], we define our closure cost to be $C(\boldsymbol{X}) = \frac{G(\boldsymbol{X})}{A(\boldsymbol{X})}$, where $G(\boldsymbol{X})$ is the boundary gap along the perimeter of (the "on" superpixels of) $\boldsymbol{X}$, and $A(\boldsymbol{X})$ is its area. Boundary gap is a measure of the disagreement between the boundary of $\boldsymbol{X}$ and is defined to be $G(\boldsymbol{X}) = P(\boldsymbol{X}) - E(\boldsymbol{X})$, where $P(\boldsymbol{X})$ is the perimeter of $\boldsymbol{X}$ and $E(\boldsymbol{X})$ is the "edginess" of the boundary of $\boldsymbol{X}$. Out of the total number of pixels along the boundary of $\boldsymbol{X}$, $P(\boldsymbol{X})$, edginess is the number of edge pixels, with the edginess of image boundary pixels defined to be 0.

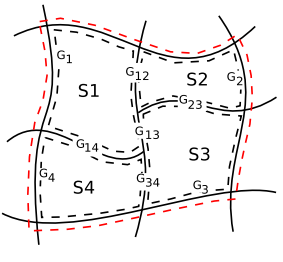


Fig. 2. Boundary gap computation over superpixel graph. S_1, S_2, S_3, and S_4 correspond to superpixels that were selected. G_i and G_{ij} are the boundary gap of superpixel i and the gap on the edge between superpixels i and j respectively . The gap along the boundary of the selection (red) is then $G_{1234} = \sum_{i=1}^{4} G_i - 2\,(G_{12} + G_{13} + G_{14} + G_{23} + G_{34})$.

In order to facilitate the optimization of this cost using an optimal graph cut-based approach (see Sec. 5), we must decompose the cost function into unary and pairwise terms of the variables in X. Let P_i be the perimeter length of

[1] The contour image takes the form of a globalPb image [24].

superpixel i and let P_{ij} be the length of the shared edge between superpixels i and j. Similarly, let E_i be the edginess of superpixel i's boundary, and E_{ij} be the edginess for the shared boundary between superpixels. Let the superpixel and superpixel edge gaps be $G_i = P_i - E_i$ and $G_{ij} = P_{ij} - E_{ij}$ respectively. Finally, let A_i be the area of superpixel i. Our closure cost becomes:

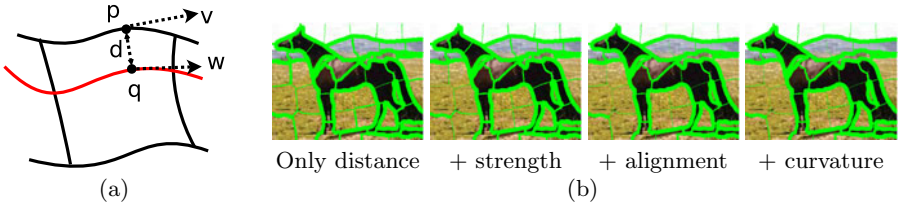
$$C(\boldsymbol{X}) = \frac{\sum_i G_i X_i - 2\sum_{i<j} G_{ij} X_i X_j}{\sum_i A_i X_i} \tag{1}$$

The denominator in the above ratio simply adds the individual areas of all the superpixels that were selected. Normalization by area not only promotes spatial coherence but also promotes compactness; as we shall see in Sec. 6, given two possible paths (with strong edge support) a closure may take, it will prefer a compact path over one with deep concavities. The numerator in the above cost is more complicated. To compute the gap along the perimeter, we first add the individual gaps of all the selected superpixels. However, for selected superpixels that share boundaries, adding individual superpixel gaps would add gaps that are not on the boundary of the selection. For every internal boundary, the gap over that boundary was counted twice (once for each of the superpixels that share the boundary). Therefore, we subtract the gap twice for all internal boundaries. Note that if two superpixels do not have a shared boundary, then both P_{ij} and E_{ij} (and thus G_{ij}) will be 0. Fig. 2 gives an example of gap computation over a simple superpixel graph. In the next section, we introduce our gap measure, and show how it can be learned from training data.

4 Learning the Gap Measure

Most approaches to detecting contour closure (e.g., [4]) typically define gap as simply the length of the missing contour fragments, i.e., the length of that portion of the closure for which no image edges exist. In order to ground our gap measure using image evidence, as well as incorporate multiple contour features for gap computation, we choose to learn the gap from ground truth. Remember from Sec. 3 that for a pair of superpixels i and j, the gap on the edge between them is $G_{ij} = P_{ij} - E_{ij}$. Specifically, if $\boldsymbol{EP_{ij}}$ is the set of pixels on the superpixel edge (i, j), then $P_{ij} = |\boldsymbol{EP_{ij}}|$ and $E_{ij} = \sum_{p \in \boldsymbol{EP_{ij}}} E_{ij}^p$, where $E_{ij}^p = [P(\boldsymbol{f^p}) > T_e]$ is an edge indicator for pixel p ($P(\cdot)$ is a logistic regressor and $\boldsymbol{f^p}$ is a feature vector for the pixel p). T_e is a necessary threshold on the edginess measure. Since the distribution of edges in the training set is not necessarily the same as that for test images, this parameter controls the contribution of weak edges. Decreasing it results in many smaller structures being detected and causes more potential solutions to be generated. We analyze the performance of our method as a function of this parameter in Sec. 6.

Given a pixel p on the superpixel boundary, the feature vector $\boldsymbol{f^p}$ is a function of both the local geometry of the superpixel boundary and the detected image edge evidence in the neighborhood of the superpixel boundary pixel. This feature vector consists of four features (see Fig. 3(a)):



(a) (b)

Fig. 3. (a) Contour features for learning the gap measure. Black curves correspond to superpixel boundaries, while the red curve corresponds to detected image edges. The features that are used for edge probability computation at superpixel boundary pixel p are: 1) distance d between p and q, where q is the closest point to p among the detected image edges; 2) image edge strength at q; 3) the alignment, computed as the absolute value of the cosine of the angle between v and w; and 4) the smoothness, computed as the squared curvature at p. (b) Effect of different features on gap (ordered left to right). For example, the superpixel edges that cross the legs weaken as alignment is added and the shadow edge on the body weakens as strength is added.

1. Distance to the nearest image edge; closer edges provide stronger evidence.
2. Strength of the nearest image edge; stronger edges provide stronger evidence.
3. Alignment between the tangent to the superpixel boundary pixel and the tangent to the nearest image edge; aligned edges provide stronger evidence.
4. Squared curvature of the superpixel edge at a point.

Given a dataset of images with manually labeled figure/ground, we map the ground truth onto superpixels. Our training set is composed of all the pixels falling on superpixel boundaries and is used to train a logistic classifier over a feature vector $\boldsymbol{f^p}$. In addition to learning from all four of the above features, we tried learning from subsets of the features. Fig. 3(b) illustrates the effect of incrementally adding more features; the thickness of each superpixel edge corresponds to the average edge probability of its superpixel boundary pixels. Using all four features results in the best performance, in terms of retaining object boundary edges while suppressing other edges.

5 Optimization Framework

It has been known for some time that ratios of real variables that adhere to certain constraints can be minimized globally. Instead of minimizing the ratio $R(x) = \frac{P(x)}{Q(x)}$ directly, one can minimize a parametrized difference $E(x, \lambda) = P(x) - \lambda Q(x)$. It can be shown that the optimal λ corresponds to the optimal ratio $\frac{P(x)}{Q(x)}$. The constraints on the ratio guarantee that the resulting difference is concave and thus can be minimized globally.

In the case of binary variables, ratio minimization can be reduced to solving a parametric maxflow problem. Kolmogorov et al. [5] showed that under certain constraints on the ratio $R(x)$, the energy $E(x, \lambda)$ is submodular and can thus be

minimized globally in polynomial time using min-cuts. Converting our closure cost $C(\boldsymbol{X})$ in Eqn. 1 to a parametrized difference results in a submodular cost $C(\boldsymbol{X}, \lambda)$, making the method in [5] applicable for minimizing the ratio $C(\boldsymbol{X})$.

In fact, the method in [5] does not simply optimize the ratio $R(x)$, but finds all intervals of λ (and the corresponding x) for which the solution x remains constant. The interval boundaries are called breakpoints, and while the smallest breakpoint λ_0 corresponds to the optimal ratio $R(x)$, consecutively larger breakpoints $\lambda_1, \lambda_2, \ldots$ are also related to ratio optimization. Kolmogorov et al. show that the optimal solution x^* of $E(x, \lambda)$ in the interval $[\lambda_i, \lambda_{i+1}]$, is also an optimal solution of $\min_{Q(x) \geq T} R(x)$, where $T = Q(x^*)$. In case of optimizing the closure cost in Eqn. 1, using parametric maxflow results in a multiscale set of optimal closure solutions under increasing area thresholds.

The method in [5] can be exponential if the number of breakpoints is exponential, but is polynomial for obtaining a global optimum. In our experiments, a solution is obtained in a fraction of a second for a superpixel graph of 200 superpixels, as there are typically less than 10 breakpoints.

6 Evaluation

We compare our work, which we refer to as *superpixel closure* (SC), to two other contour grouping methods: Estrada and Jepson (EJ) [3] and a version of ratio contours (RRC) from Stahl and Wang [4]. We provide a qualitative evaluation on various images (see Fig. 6), as well as a quantitative evaluation on two datasets, including the Weizmann Horse Database (WHD) [27] and the Weizmann Segmentation Database (WSD) [28]. Learning the gap measure (Sec. 4) is accomplished on the first 30 images from WHD. For testing, we use 170 additional images from WHD and all 100 images from WSD.

6.1 Quantitative Evaluation

For a quantitative evaluation of the results, we use the **F-measure**, $F = \frac{2RP}{R+P}$, where R and P are recall and precision, respectively, of the solution relative to the ground truth. Specifically, if A is the set of pixels corresponding to the solution and A_{gt} is the ground truth, then $R = \frac{|A \cap A_{gt}|}{|A_{gt}|}$ and $P = \frac{|A \cap A_{gt}|}{|A|}$. Given K solutions, we select the solution with the best F-measure relative to the ground truth. We average the "per-image" F-measure for all the images (and three ground truth segmentations in WSD) in a dataset and report the result.

Fig. 4 shows the results of the three methods for increasing values of K. We chose the best parameters for all three algorithms and fixed them for the entire experiment. For EJ, we used a Normalized Affinity Threshold ($\tau_{afft y}$) of 0.01, with the line segments generated by fitting the globalPb output. For RRC, we used $\lambda = 0$ and $\alpha = 1$. Here, we could not give the algorithm globalPb-based line segments, and thus use the method's own line segments generated from a Canny edge response. For our method, we fixed the number of superpixels to 200 and set $T_e = 0.05$, giving us best performance at the high range of K.

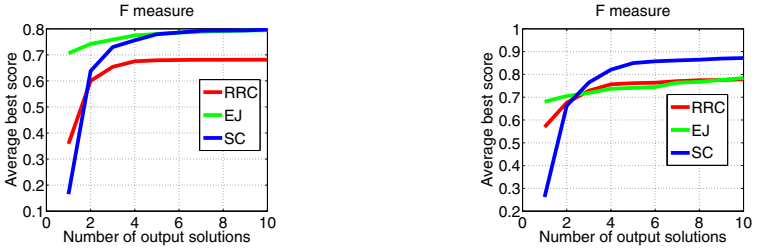


Fig. 4. Quantitative results. We compare our results (SC) to two other algorithms: Estrada and Jepson [3] (EJ) and Ratio Contours [4] (RRC).

Since the resulting solutions can be thought of as shape hypotheses for object recognition, we believe that the performance for some reasonably small value of $K > 1$ is more important than aiming to obtain a single best contour ($K = 1$)[2]. For $K = 10$, SC (EJ, RRC) obtains an average F-measure of 79.72% (79.44%, 68.13%) on WHD and 87.19%[3] (78.44%, 77.82%) on WSD.

We outperform the competing approaches on both datasets for a setting of $K = 10$ (obtaining a comparable performance to EJ on the horses dataset), which we attribute to the superpixel formulation, as well as the optimal closure finding method in our framework. On the WHD, both SC and RRC perform significantly worse than on WSD, while EJ performs the same. This is likely due to the lower compactness of objects in the horse dataset (average isoperimetric ratio of 0.15, compared to 0.4 in WSD). Moreover, in many images there is a more compact path that includes the gap between the horse's legs due to shadow or ground edges. In addition, a significant number of images in the horse dataset have a picture frame boundary around the image. These boundaries provide the largest and most compact solutions, and are therefore found by SC instead of finding the horse. Finally, an intersting fact is that EJ still performs well on the horse dataset (unlike SC and RRC). This is most likely due to its reliance on internal appearance, which is definetely homogeneous in the case of horses.

Since T_e is set so low, our performance is poor for low values of K, but it is better for high K's. Fig. 5 shows the change in performance of our algorithm as we change the number of superpixels and vary T_e. Fig. 5(a) illustrates that, in general, higher superpixel density results in a very marginal performance gain for large values of K, while for low values of K, coarser superpixel segmentations prevent very small objects from being detected. Increasing the threshold T_e (Fig. 5(b)) reduces the detection of small objects and improves performance at

[2] SC can be tuned (see Fig. 5) to perform better for $K = 1$ at a small expense of performance for higher K's.

[3] For WSD, there are three ground truth segmentations per image. If we instead choose the closest of the three ground truth segmentations per image (as opposed to taking the average), our score on WSD improves to 88.76%.

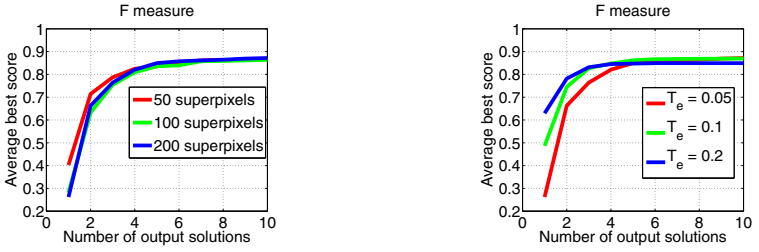


Fig. 5. Varying the parameters of our method. (a) Varying the number of superpixels for a fixed edge threshold $T_e = 0.05$. (b) Varying the edge threshold T_e for a fixed number of superpixels (200).

the low range of K, but also hurts the detection of objects with weak edges and thus results in slightly poorer performance at the high range of K.

We also compare the running times of the three methods on WSD (average image size of 300×290 pixels). On a 2.6GHz Dual Core Intel CPU with 4GB of RAM, setting the methods to retrieve $K = 10$ best contours, the average running times per image are: SC (not including edge detection and superpixel extraction) – 1.3 sec, EJ (not including edge detection) – 23 sec, RRC – 59 sec.

6.2 Qualitative Evaluation

In addition to the quantitative evaluation, we also provide a qualitative evaluation of our method by testing it on images from the two datasets, as well as other images obtained from the internet. Fig. 6 illustrates the performance of our method compared to the two competing approaches[4]. We manually select the best of 10 solutions for each method. Notice that the detected contours in our framework lie closer to the true object contours since the superpixel edges, even in the persence of a gap, lie closer to object edges than the linearized contours detected by the other algorithms. We pleasantly observed that our framework is not constrained to obtain compact solutions as is usually the case when one is normalizing perimeter by area. This is clearly visible in the image of a spider, where very thin legs are segmented since that represents the best closure solution. However, this is not always the case, for if there is a more compact contour that is not losing on gap, it will be preferred. This is the reason for the filled gap between the horse's legs or the filled gap between the carriage's wheels in the first two images. Note that for the horse image, EJ obtains a better solution by relying on the homogeneous appearance inside the horse. Finally, our method relies on superpixels to oversegment the object. We still detect thin structures, such as the spider's legs, if good superpixels were found due to strong image edges. For weaker edges, however, thin structures are harder to capture (the bat of the baseball player, for example).

[4] Supplementary material contains the results of SC for all the images in both datasets.

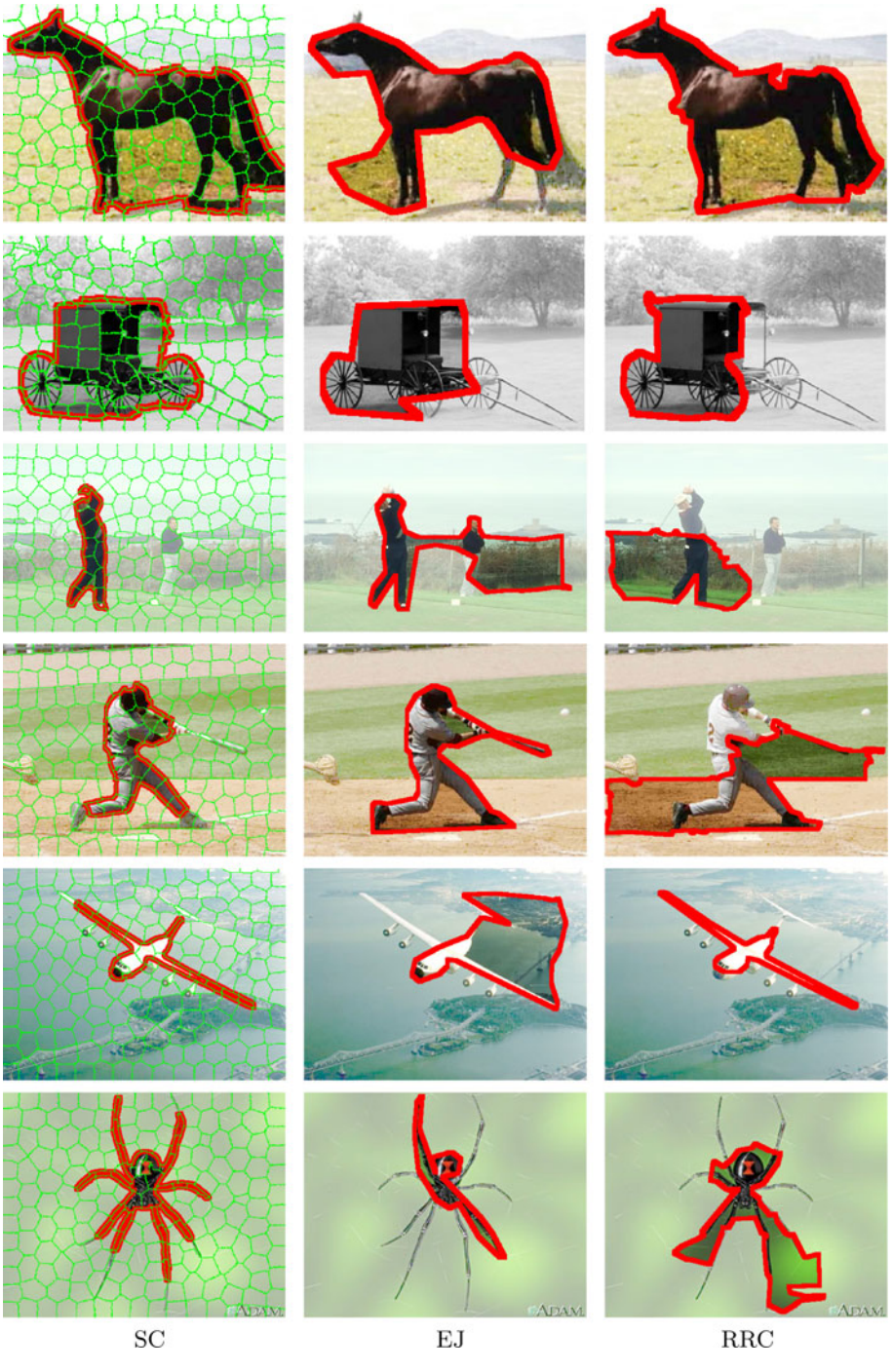


Fig. 6. Qualitative results. We compare our results (left) to two other algorithms: Estrada and Jepson [3] (middle) and Ratio Contours [4] (right).

6.3 Using Internal Homogeneity

As mentioned in Sec. 1, our superpixel formulation also facilitates the incorporation of appearance information, when it is both available and appropriate. The cost function in Eqn. 1 can be easily modified to incorporate a term which reflects the degree to which adjacent superpixels *inside* the superpixel selection, i.e., inside the closed contour, have high affinity. Assuming that we are given an affinity matrix W, such that W_{ij} is the affinity between two superpixels i and j, we modify our closure cost to be:

$$C_{affty}(\boldsymbol{X}) = \frac{\sum_i G_i X_i - 2\sum_{i<j} G_{ij} X_i X_j}{\sum_{i<j} W_{ij} X_i X_j} \tag{2}$$

Compared to the cost in Eqn. 1, the numerator remains the same while the denominator changes to an internal homogeneity measure instead of the total object area. Minimizing this ratio results in minimizing the gap while maximizing the total affinity between the selected superpixels. Fig. 7 shows an example where better results were achieved by exploiting appearance homogeneity.

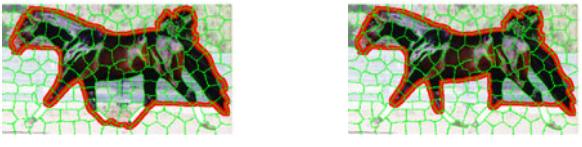


Fig. 7. Using internal appearance homogeneity. For objects with strong internal homogeneity of appearance, optimizing the cost in Eqn. 2 is better (right) than optimizing the cost in Eqn. 1 (left). Note that the gap between the horse's legs was not included on the right due to its hetrogeneous appearance w.r.t. the rest of the horse.

6.4 Multiple Superpixel Scales

Though it might seem that the more superpixels we use, the better SC will perform, it is not always so. As seen in Fig. 5(a), coarser superpixel scales constrain the solution more and thus perform better for low values of K. However, there is one additional advantage of using coarser superpixel scales. Since our superpixel algorithm does not produce hierarchical superpixels (since new superpixel boundaries may be introduced from finer to coarser scales), it is possible to occasionally have less undersegmentation at coarser scales. Fig. 8 illustrates a situation where an object was segmented better at a coarser scale and consequently detected by SC.

We tried a simple multiscale version of SC where we merge the results from all scales. Specifically, we run SC at four superpixel scales, obtaining 25, 50, 100, and 200 superpixels for each image. Setting $K = 10$ for each scale results in 40 solutions once the results are merged together. Since the performance of SC for a given scale does not significantly vary for $K > 10$, we do not select 10 of 40 solutions for the multiscale version, but instead retain all 40. Using the multiscale version increases the performance on WSD from 87.19% to 89.53%.

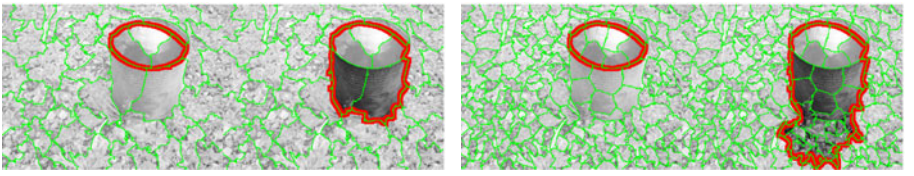


Fig. 8. Multiscale results. Choosing the $K = 2$ top solutions yields better results in the case of 50 superpixels (left) than in the case of 200 superpixels (right).

7 Conclusions

Our reformulation of the problem of finding cycles of contours as the problem of finding spatially coherent subsets of superpixels, whose collective boundary has strong image edge evidence yields an optimal framework for closure detection that compares favorably with two leading prior approaches. While superpixels provide an ideal scope for learning a gap measure from training data, they offer a number of additional advantages that we are currently exploring. We plan to use superpixel junctions to learn an affinity measure between pairs of superpixels that are both inside and adjacent to the boundary. Such an affinity measure can encode a learned measure of continuity and T-junction, and could significantly strengthen our cost function. Superpixels also provide a convenient mechanism for incorporating appearance information, if appropriate and if available. For example, if the object was known to be homogeneous in appearance, our modified cost function can easily incorporate such a prior, as discussed in Sec. 6.3. Our framework is flexible, and can easily accommodate many classical non-accidental regularities. In the future, we also plan to pursue a more elegant coarse-to-fine framework for finding contour closure using multiple superpixel scales.

Acknowledgements

We thank Allan Jepson for discussion about closure cost functions and optimization procedures, and Yuri Boykov and Vladimir Kolmogorov for providing their parametric maxflow implementation. This work was supported in part by the European Commission under a Marie Curie Excellence Grant MCEXT-025481 (Cristian Sminchisescu) and NSERC (Alex Levinshtein, Sven Dickinson).

References

1. Wertheimer, M.: Laws of organization in perceptual forms. In: Ellis, W. (ed.) Source Book of Gestalt Psychology. Harcourt, Brace (1938)
2. Estrada, F.J., Jepson, A.D.: Perceptual grouping for contour extraction. In: ICPR, pp. 32–35 (2004)
3. Estrada, F.J., Jepson, A.D.: Robust boundary detection with adaptive grouping. In: POCV, p. 184 (2006)

4. Stahl, J., Wang, S.: Edge grouping combining boundary and region information. IEEE Transactions on Image Processing 16, 2590–2606 (2007)
5. Kolmogorov, V., Boykov, Y., Rother, C.: Applications of parametric maxflow in computer vision. In: ICCV (2007)
6. Carreira, J., Sminchisescu, C.: Constrained parametric min-cuts for automatic object segmentation. In: CVPR (2010)
7. Ren, X., Fowlkes, C.C., Malik, J.: Cue integration in figure/ground labeling. In: NIPS, pp. 1121–1128 (2005)
8. Ren, X., Fowlkes, C.C., Malik, J.: Scale-invariant contour completion using conditional random fields. In: ICCV, pp. 1214–1221 (2005)
9. Williams, L.R., Jacobs, D.W.: Stochastic completion fields: a neural model of illusory contour shape and salience. In: ICCV, p. 408 (1995)
10. Lowe, D.G.: Perceptual Organization and Visual Recognition. Kluwer Academic Publishers, Norwell (1985)
11. Brady, M., Asada, H.: Smoothed local symmetries and their implementation. IJRR 3, 36–61 (1984)
12. Cham, T.J., Cipolla, R.: Geometric saliency of curve correspondances and grouping of symmetric comntours. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065, pp. 385–398. Springer, Heidelberg (1996)
13. Saint-Marc, P., Rom, H., Medioni, G.: B-spline contour representation and symmetry detection. PAMI 15, 1191–1197 (1993)
14. Ylä-Jääski, A., Ade, F.: Grouping symmetrical structures for object segmentation and description. CVIU 63, 399–417 (1996)
15. Stahl, J.S., Wang, S.: Globally optimal grouping for symmetric closed boundaries by combining boundary and region information. PAMI 30, 395–411 (2008)
16. Levinshtein, A., Dickinson, S., Sminchisescu, C.: Multiscale Symmetric Part Detection and Grouping. In: ICCV (2009)
17. Jacobs, D.W.: Robust and efficient detection of salient convex groups. PAMI 18, 23–37 (1996)
18. Elder, J.H., Zucker, S.W.: Computing contour closure. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065, pp. 399–412. Springer, Heidelberg (1996)
19. Wang, S., Kubota, T., Siskind, J.M., Wang, J.: Salient closed boundary extraction with ratio contour. PAMI 27, 546–561 (2005)
20. Elder, J., Zucker, S.: A measure of closure. Vision Research 34, 3361–3369 (1994)
21. Williams, L.R., Hanson, A.R.: Perceptual completion of occluded surfaces. CVIU 64, 1–20 (1996)
22. Zhu, Q., Song, G., Shi, J.: Untangling cycles for contour grouping. In: ICCV (2007)
23. Jermyn, I., Ishikawa, H.: Globally optimal regions and boundaries as minimum ratio weight cycles. PAMI 23, 1075–1088 (2001)
24. Maire, M., Arbelaez, P., Fowlkes, C., Malik, J.: Using contours to detect and localize junctions in natural images. In: CVPR (2008)
25. Mori, G., Ren, X., Efros, A.A., Malik, J.: Recovering human body configurations: Combining segmentation and recognition. In: CVPR, pp. 326–333 (2004)
26. Martin, D.R., Fowlkes, C.C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. PAMI 26, 530–549 (2004)
27. Borenstein, E., Ullman, S.: Class-specific, top-down segmentation. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2351, pp. 109–124. Springer, Heidelberg (2002)
28. Alpert, S., Galun, M., Basri, R., Brandt, A.: Image segmentation by probabilistic bottom-up aggregation and cue integration. In: CVPR (2007)

Fast and Exact Primal-Dual Iterations for Variational Problems in Computer Vision

Jan Lellmann, Dirk Breitenreicher, and Christoph Schnörr

Image and Pattern Analysis Group & HCI
Dept. of Mathematics and Computer Science, University of Heidelberg
{lellmann,breitenreicher,schnoerr}@math.uni-heidelberg.de

Abstract. The saddle point framework provides a convenient way to formulate many convex variational problems that occur in computer vision. The framework unifies a broad range of data and regularization terms, and is particularly suited for nonsmooth problems such as Total Variation-based approaches to image labeling. However, for many interesting problems the constraint sets involved are difficult to handle numerically. State-of-the-art methods rely on using nested iterative projections, which induces both theoretical and practical convergence issues. We present a dual multiple-constraint Douglas-Rachford splitting approach that is globally convergent, avoids inner iterative loops, enforces the constraints exactly, and requires only basic operations that can be easily parallelized. The method outperforms existing methods by a factor of $4-20$ while considerably increasing the numerical robustness.

1 Introduction

Overview and Motivation. In this work, we focus on algorithms for solving saddle point problems associated with variational formulations in image processing and analysis, which have recently become a very active research area. The output of a variational method is defined as the minimizer

$$u^* := \arg\min_{u \in \mathcal{C}} f(u)\,, \tag{1}$$

where $\mathcal{C}$ is some subset of a space of functions defined on some continuous domain, and f a functional depending on the input data. In contrast to "discretize first" approaches such as grid- or graph based methods, this "analyze first" approach allows to get a deeper insight into the underlying problem, and to abstract from inaccuracies caused by the discretization.

The interpretation of u is governed by the problem to be solved: for color denoising, $u : \Omega \to [0,1]^3$ could directly describe the colors of the output image on the image domain $\Omega \subseteq \mathbb{R}^d$; while for segmentation problems, $u : \Omega \to [0,1]$ could assign each point to the foreground ($u(x) = 1$) or background ($u(x) = 0$) class. Recently, interest has risen in a specific class of variational problems of the form

$$\inf_{u \in \mathcal{C}} \sup_{v \in \mathcal{D}} \left\{ \langle u, s \rangle + \langle Lu, v \rangle - \langle b, v \rangle \right\}\,, \tag{2}$$

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 494–505, 2010.

Fig. 1. Application of the proposed saddle point optimization method to multi-class color segmentation. **Left:** Input image. **Right:** Segmentation into 12 regions of constant color. The tight relaxation of the combinatorial labeling problem results in a saddle point problem with an intricate dual constraint set. In contrast to existing approaches, the method proposed in this work allows to compute global minimizers of such problems without requiring inaccurate and time-consuming iterative projections as subroutines.

where the *primal* and *dual constraint sets* $\mathcal{C} \subseteq X$ and $\mathcal{D} \subseteq Y$ are convex subsets of some function space X with dual space Y, $L : X \to Y$ is a linear operator, $s \in Y$ and $b \in X$. These *bilinear saddle point problems* are very useful in the context of labeling [4,12,14], and – using a "lifting" technique – can be used to minimize a large class of common variational problems [18].

As these problems are generally convex, they do not suffer from local minima, which allows to clearly separate modelling from optimization aspects. The inner problem turns out to be a convenient way of expressing objective functions f that contain *non-smooth* terms, such as Total Variation (TV) regularization, and allows to apply fast *primal-dual* optimization schemes that explicitly update the primal variables u as well as the dual variables v.

First-order methods of this kind have been shown to achieve a good performance for many problems while offering excellent parallelization characteristics [22,17,14]. These methods require to compute projections $\Pi_{\mathcal{C}}$ and $\Pi_{\mathcal{D}}$ on the sets $\mathcal{C}$ and $\mathcal{D}$. However, in many cases one faces discretized problems of the form

$$\min_{u \in \mathcal{C}} \max_{v \in \mathcal{D}_1 \cap \ldots \cap \mathcal{D}_r} \{\langle u, s\rangle + \langle Lu, v\rangle - \langle b, v\rangle\} \;, \tag{3}$$

with $\mathcal{C} \subseteq \mathbb{R}^n$ and $\mathcal{D}_i \subseteq \mathbb{R}^m, i = 1, \ldots, r$. This occurs in particular in connection with relaxations of the combinatorial labeling problem (Fig. 1) and functional lifting [4,18,14]. Here the dual constraint set $\mathcal{D}$ is only given implicitly as an intersection, hence projections cannot be computed in closed form.

Current methods to solve such problems are based on *approximating* the projection on $\mathcal{D}$ by a series of projections on the simpler sets $\mathcal{D}_i$. However, this causes a number of issues. From a theoretical viewpoint, convergence of the outer algorithms usually requires the inner problem to be solved with an increasing accuracy at each step, which is impractical. Thus in practice convergence is no

longer guaranteed. In addition, the projections become very slow, and raise many issues on how to choose suitable and matching stopping criteria.

Contribution. In this work, we propose a dual multiple-constraint Douglas-Rachford method for saddle point problems of the class (3), that *exactly* takes into account the dual constraint set $\mathcal{D}$ while still relying only on simple exact operations. The method is shown to converge to a global optimum and is suited for massive parallelization. While the method essentially solves the dual problem, we show that a primal solution can be recovered. As all steps in the proposed algorithm can be computed explicitly, the theoretical convergence results directly transfer to the actual implementation. The approach outperforms state-of-the-art methods on real-world problems with respect to computation time and numerical robustness by a factor of $4 - 20$.

Related Work. Continuous labeling approaches [21,5] constitute a continuous equivalent to discrete graph cut methods [3]. These discrete methods are difficult to parallelize and suffer from anisotropy induced by the discretization. This *grid bias* can be reduced in some extent by using larger neighborhoods in the graph construction, but it cannot be completely eliminated and computational costs quickly increase in the process. In contrast, continuous methods can be used to construct discretizations that exactly represent the original metric in an infinitesimal sense [4]. The idea of functional lifting can be found in a discrete setting in [11] and in a continuous formulation in [4,17], and has also proven to be useful in the context of optical flow [10].

Regarding optimization, our work extends the approach proposed in [9] for two-, and in [14] for multiclass labeling. The authors use a similar method, but require iterative projections at each step. The basic Douglas-Rachford iteration [6,7] applied to the dual problem can be shown to be equivalent to the Alternating Direction Method of Multipliers [8] and the recently proposed Alternating Split Bregman method [9,20], hence our results equally apply in these formulations.

2 Bilinear Saddle-Point Problems in Computer Vision

In the following, we will consider variational problems that can be stated in the specific saddle point form (3) when discretized. For $s \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $L \in \mathbb{R}^{m \times n}$, and some closed convex sets $\mathcal{C} \subseteq \mathbb{R}^n$ and $\mathcal{D}_i \subseteq \mathbb{R}^m, i = 1, \dots, r$, define $\mathcal{D} := \mathcal{D}_1 \cap \dots \cap \mathcal{D}_r$ and

$$g(u, v) := \langle u, s \rangle + \langle Lu, v \rangle - \langle b, v \rangle \,. \tag{4}$$

Then problem (3) consists in computing a minimizer of the *primal objective* $f(u) := \max_{v \in \mathcal{D}} g(u, v)$,

$$\min_{u \in \mathcal{C}} \max_{v \in \mathcal{D}} g(u, v) = \min_{u \in \mathcal{C}} f(u) \,. \tag{5}$$

Under the assumption that at least one of the sets $\mathcal{C}$ and $\mathcal{D}$ is bounded, it can be shown that equivalently one may maximize the *dual objective* $f_d(v) := \min_{u \in \mathcal{C}} g(u, v)$ [19, Cor. 37.3.2],

$$\max_{v\in\mathcal{D}} \min_{u\in\mathcal{C}} g(u,v) = \max_{v\in\mathcal{D}} f_d(v)\,. \tag{6}$$

In particular, pairs of primal resp. dual solutions (u^*, v^*) are *saddle points* of g,

$$\min_{u\in\mathcal{C}} f(u) = f(u^*) = g(u^*,v^*) = f_d(v^*) = \max_{v\in\mathcal{D}} f_d(v) \tag{7}$$

We will now present two prototypical applications of the saddle point method: multiclass image labeling and generic scalar variational models with gradient-based regularizers.

Continuous Multiclass Labeling Approaches. Many problems in image analysis can be reduced to the basic problem of assigning to each point x in the image domain Ω one of l discrete labels $\{1,\dots,l\}$, such as an object class in segmentation problems, a depth label in stereo reconstruction, or a displacement vector in image registration [16]. In order to reduce the influence of noise, some nonlocal spatial coherency constraints are required in addition to the local data fidelity measure based on the input image.

As for each point a discrete decision must be made, the problem is combinatorial and nonconvex, and in fact can be shown to be NP-hard even for relatively simple energies under a graph discretization [3]. However, by relaxing the original problem to a convex constraint set, good solutions for the original problem can be recovered using convex optimization [22,4,13,14]. In the continuous setting, the labeling problem can be relaxed to the variational problem

$$\min_{u\in\mathcal{C}} \langle u, s\rangle + J(u)\,, \quad \mathcal{C} := \{u\in \mathrm{BV}(\Omega,\mathbb{R}^l) | u(x) \geqslant 0, \sum_i u_i(x) = 1\}\,, \tag{8}$$

where BV denotes the space of functions of bounded variation [2]. By embedding the original labels into a higher-dimensional space via the unit vectors $\{e^1,\dots,e^l\}$, the local data fidelity can be completely encoded into the linear term, irrespective of the complexity of the original data term: assigning label i to the point x will locally be penalized by $s_i(x)$.

For the nonlocal regularizer J, we choose some metric $d : \{1,\dots,l\}^2 \to \mathbb{R}$, denote by Du the (distributional) Jacobian of u, and set

$$J(u) := \sup_{v\in\mathcal{D}} \int_\Omega \langle Du, v\rangle, \quad \mathcal{D} := \{v\in (C_c^\infty)^{d\times l} | v(x)\in\mathcal{D}_{\mathrm{loc}} \forall x\in\Omega\}\,, \tag{9}$$

$$\mathcal{D}_{\mathrm{loc}} := \{v = (v^1,\dots,v^l)\in\mathbb{R}^{d\times l} | \|v^i - v^j\| \leqslant d(i,j), \sum_k v^k = 0\}\,. \tag{10}$$

This is a tight relaxation of the requirement that switching from label i to label j along some curve should be penalized by the curve length, multiplied by a factor $d(i,j)$ *depending on the labels* i *and* j. In terms of graph-based approaches, this can be thought of as the potentials on the edges of the graph. The formulation (9) carries over this principle to the continuous domain Ω. By discretizing u, v and s on a rectangular grid and choosing a forward finite differences discretization L of the gradient operator D, the above variational formulation can be posed in the saddle point form (3) without introducing grid bias (cf. [4]).

The definition of $\mathcal{D}_{\text{loc}}$ is derived by locally constructing the convex envelope of the desired regularizer restricted to the set of u that only assume the "hard" labels $\{e^1, \dots, e^l\}$. As a result, the minimizer of the *convex* problem (8) is often a unit vector in almost all points, and provides a very good approximation to the solution of the original *combinatorial* labeling problem.

The increased *approximation tightness* comes at the price of a more complicated *optimization* process. However, as $\mathcal{D}_{\text{loc}}$ is the intersection

$$\mathcal{D}_{\text{loc}} = \{v \in \mathbb{R}^{d\times l} | \sum_i v^i = 0\} \cap \bigcap_{i<j} \{v \in \mathbb{R}^{d\times l} | \|v^i - v^j\| \leqslant d(i,j)\}\,, \tag{11}$$

the problem can be put into the form (3). Projections on the *individual* sets can be easily computed by subtracting the mean resp. by shrinkage-like operations.

Lifting Approach. For the case where the sought-after function takes scalar values, such as gray scale or depth, the saddle point formulation permits another interesting application. Assume we want to minimize over $\mathcal{C} \subseteq W^{1,1}(\Omega, \mathbb{R})$ some functional

$$\min_{u' \in \mathcal{C}} f'(u'), \quad f'(u') := \int_\Omega h(x, u'(x), \nabla u'(x)) dx \tag{12}$$

with h convex in $\nabla u'(x)$, but not necessarily in $u'(x)$. Then, motivated by the "calibration" idea [1], it was shown in [18] that f' can be expressed in terms of the $\{0,1\}$-indicator function $\chi_{H(u')}$ of the hypograph

$$H(u') := \big\{(x,t) \in \mathbb{R}^d \times \mathbb{R} | u'(x) \geqslant t\big\} \tag{13}$$

of u', i.e. $\chi_{H(u')}(x,t) = 1$ iff $u'(x) \geqslant t$ and zero otherwise. Specifically,

$$f'(u') = \sup_{v \in \mathcal{D}} \int_{\Omega\times\mathbb{R}} \langle v, D\chi_{H(u')} \rangle\,, \quad \text{where} \tag{14}$$

$$\mathcal{D} := \big\{(v^x, v^t) \in \mathcal{C}_c^\infty(\Omega \times \mathbb{R}, \mathbb{R}^{d+1}) | \forall x, t : v^t(x,t) \geqslant h^*(x,t,v^x(x,t))\big\}\,. \tag{15}$$

Here h^* denotes the convex conjugate of h with respect to the last argument. Intuitively, this *lifts* the problem to a higher-dimensional space and transforms it to the problem of finding the *set of points below the graph* of u'.

Again, the problem is transformed to a convex problem by replacing $\chi_{H(u')}$ with some function $u : \Omega\times\mathbb{R} \to [0,1]$. This effectively linearizes the nonconvexity of h with respect to $u'(x)$. The relaxed problem reads

$$\min_{u\in\mathcal{C}} \sup_{v\in\mathcal{D}} \int_{\Omega\times\mathbb{R}} \langle v, Du \rangle\,, \quad \mathcal{C} := \{u \in \mathrm{BV}(\Omega\times\mathbb{R}, [0,1]) | u(x,t) \overset{t\to\pm\infty}{\longrightarrow} 0/1\}\,, \tag{16}$$

which after discretization fits into the saddle point framework (3). Again, depending on the *integrand* h, the *dual constraint set* $\mathcal{D}$ may be very complicated. The approach can be extended to the full Mumford-Shah functional [15],

$$f'(u') = \lambda \int_\Omega (u' - I)^2 dx + \int_{\Omega\setminus S_{u'}} \|\nabla u'\|^2 dx + \nu \mathcal{H}^{d-1}(S_{u'}), \quad \lambda, \nu > 0\,, \tag{17}$$

Fig. 2. Application of the proposed optimization method to nonsmooth variational denoising. **Left:** Input image. **Right:** Result of variational denoising using the Mumford-Shah functional with 8 levels, $\lambda = 0.5$ and $\nu = 5$. Noise or fine details can be removed without blurring sharp edges. The lifting approach allows to minimize the full Mumford-Shah functional within the convex saddle point framework.

where $\mathcal{H}^{d-1}$ is the $(d-1)$-dimensional Hausdorff measure (Fig. 2). The $W^{1,1}$ requirement above is relaxed to $u' \in \mathrm{SBV}(\Omega \times \mathbb{R})$, i.e. the set of special functions of bounded variation [2], such that u' may have a nonempty set of discontinuities $S_{u'}$. The dual constraint set then becomes [17]

$$\mathcal{D} = C_c^\infty(\Omega \times \mathbb{R}, \mathbb{R}^{d+1}) \cap \mathcal{R} \cap \bigcap_{p \leqslant q} \mathcal{S}_{p,q}\,, \tag{18}$$

$$\mathcal{R} := \left\{ (v^x, v^t) | v^t(x,t) + \lambda(t - f(x))^2 \geqslant \frac{\|v^x(x,t)\|^2}{4} \; \forall x, t \right\} \tag{19}$$

$$\mathcal{S}_{p,q} := \left\{ (v^x, v^t) | \left\| \int_p^q v^x(x,t) dt \right\| \leqslant \nu \; \forall x, t \right\}. \tag{20}$$

Again, projections on the discrete counterpart of $\mathcal{D}$ can only be *approximated*. On the other hand, projections on $\mathcal{S}_{p,q}$ and $\mathcal{R}$ can be computed explicitly by using a shrinkage-like method [4] resp. by solving a third-order polynomial using a solution formula. This motivates our optimization approach below that *exactly* takes $\mathcal{D}$ into account in terms of individual projections onto $\mathcal{S}_{p,q}$ and $\mathcal{R}$.

3 Dual Multiple-Constraint Douglas-Rachford Splitting

Based on the theory of set-valued operators applied to the subdifferential operators of convex functions, the Douglas-Rachford approach [6] provides a scheme to compute a minimizer of the problem

$$\min_{u \in \mathbb{R}^n} f(u), \quad f(u) := f_1(u) + f_2(u)\,, \tag{21}$$

by iterating a combination of backward (proximal) steps with step size $\tau > 0$,

$$u' \leftarrow \arg\min_{u' \in \mathbb{R}^n} \{(2\tau)^{-1} \|u' - u\|_2^2 + f_i(u')\} \tag{22}$$

Algorithm 1. Dual Multiple-Constraint Douglas-Rachford Optimization for Saddle-Point Problems (DMDR)

Choose $\tau > 0, \bar{v}_i^0 \in \mathbb{R}^{n\times d\times l}, \bar{z}^0 \in \mathbb{R}^{n\times d}$. Set $k \leftarrow 0$.
while (not converged) **do**
  $v_i^k \leftarrow \Pi_{\mathcal{D}_i}\left(\bar{v}_i^k - \frac{\tau}{r} b\right)$.
  $z''^k \leftarrow \Pi_{\mathcal{C}}\left(\frac{1}{\tau}\left(\bar{z}^k - s\right)\right)$.
  $v'^k \leftarrow \left(rI + LL^\top\right)^{-1}\left(\sum_i \left(2v_i^k - \bar{v}_i^k\right) - L\left(\bar{z}^k - 2\tau z''^k\right)\right)$.
  $v_1'^k = \ldots = v_r'^k \leftarrow v'^k$.
  $z'^k \leftarrow \left(-L^\top\right) v'^k$.
  $\bar{v}_i^{k+1} \leftarrow \bar{v}_i^k + v_i'^k - v_i^k$.
  $\bar{z}^{k+1} \leftarrow z'^k + \tau z''^k$.
  $k \leftarrow k+1$.
end while

on each of the f_i *individually*. More precisely, if both f_1 and f_2 are proper, convex, and lower semicontinuous functions, and the relative interiors of their domains have a nonempty intersection, the Douglas-Rachford iteration scheme converges to a minimizer of f [7, Thm. 3.15; Prop. 3.23, 3.20, 3.19]. A strong point of the method is that it does not require any part of the objective to be smooth or finite, which allows to introduce constraints into the f_i as required.

Algorithm and Convergence. We will now show how to add auxiliary variables before splitting the objective in order to avoid the iterative projections employed in [4,18] and the associated accuracy and convergence issues. Instead of solving (5) directly, we solve the *dual* problem (6) and additionally introduce auxiliary variables z and $v_1, \ldots, v_r$, leading to the equivalent problem

$$\min_{v_i \in \mathbb{R}^m} \underbrace{\delta_{-L^\top\left(\frac{1}{r}\sum_i v_i\right)=z, v_1=\ldots=v_r}}_{f_1} + \underbrace{\sum_i \delta_{v_i\in\mathcal{D}_i} + \left\langle \tfrac{1}{r}\textstyle\sum_i v_i, b\right\rangle + \max_{u\in\mathcal{C}} \langle u, z-s\rangle}_{f_2} . \quad (23)$$

The extra constraints are represented as characteristic functions δ taking values $\{0, +\infty\}$. Applying the Douglas-Rachford method to the above splitting formulation leads to the complete algorithm as outlined in Alg. 1. Due to the auxiliary variables, the backward step for f_2 requires only separate projections on the $\mathcal{D}_i$ instead of the complete set $\mathcal{D}$. The backward step for f_1 amounts to solving a linear equation system. By the Woodbury identity, this can be transformed to

$$\left(rI + LL^\top\right)^{-1} x = r^{-1}x - r^{-1}L\left(rI + L^\top L\right)^{-1} L^\top x\,. \quad (24)$$

In all of the presented applications, L is a forward differences discretization of the gradient. Thus $LL^\top$ is the five-point Laplacian and diagonalizes with respect to the discrete cosine transform, allowing to solve (24) fast and exact using DCT and diagonal matrix multiplications. We now show convergence of Alg. 1 subject to a mild condition on the relative interiors ri of the domains.

Proposition 1. *Let $\mathcal{D}_1, \ldots, \mathcal{D}_r, \mathcal{C}$ be closed convex sets, $\mathcal{C}$ bounded such that* $\mathrm{ri}(\mathcal{D}_1) \cap \ldots \cap \mathrm{ri}(\mathcal{D}_r) \neq \emptyset$ *and* $\mathrm{ri}(\mathcal{C}) \neq \emptyset$. *Then Alg. 1 converges in* $(v_1^k, \ldots, v_r^k, z''^k)$.

Proof. As $\mathcal{C}$ is closed we have $\operatorname{ri}(\operatorname{dom} f_2) \cap \operatorname{ri}(\operatorname{dom} f_1) = \operatorname{ri}(\operatorname{dom} f_2) \cap \{v_1 = \ldots = v_r, -L^\top v_i = z\} = \{(v, \ldots, v, -L^\top v)^\top \,|\, v \in \operatorname{ri}(\mathcal{D}_1) \cap \ldots \cap \operatorname{ri}(\mathcal{D}_r)\}$. This set is nonempty by the assertion, which with the remarks at the beginning of the section implies convergence. □

Duality Properties of the Proposed Method. In particular, the convergence property of the Douglas-Rachford approach guarantees that from some point on the *constraints hold exactly*. Then $v^k := v_1^k = \ldots = v_r^k$, and v^k converges to a solution v of the dual problem (6). Unfortunately, it is nontrivial to generate a primal solution u from a single dual solution, as both the dual and the primal problem are usually not strictly convex. However, it turns out that the above algorithm additionally returns a primal solution:

Proposition 2. *Let $(v := v_1 = \ldots = v_r, z'')$ be a fixed point of Alg. 1. Then z'' is a solution of the primal problem (5).*

Proof. We will only provide a sketch the proof as it is quite technical. The point is to show that the limit (z'', v) of Alg. 1 is a saddle point of $g(u, v)$ as defined in (4), i.e.

$$g(u, \tilde{v}) \leqslant g(z'', v) \leqslant g(\tilde{u}, v) \;\forall \tilde{u} \in \mathcal{C}, \tilde{v} \in \mathcal{D}\,. \tag{25}$$

Let $\bar{z}$ and $\bar{v}_i$ be the corresponding limits from Alg. 1, and substitute $z := \bar{z} - \tau z''$. Denoting by $\partial f(x)$ the subdifferential (i.e. the set of subgradients) of f in x, from the Douglas-Rachford convergence theorem [7, Prop. 3.19], it follows that

$$\tau^{-1} \left(\bar{v}_1 - v_1, \ldots, \bar{v}_r - v_r, \bar{z} - z\right)^\top \in \partial f_2(v_1, \ldots, v_r, z)\,. \tag{26}$$

Summing up and using the definition of the algorithm leads to

$$Lz'' = \tau^{-1} \sum_i (\bar{v}_i - v_i) \in \sum_i N_{\mathcal{D}_i}(v_i) + b = N_{\mathcal{D}}(v) + b, \tag{27}$$

where $N_{\mathcal{D}}$ denotes the normal cone of the set $\mathcal{D}$ from convex analysis. On the other hand, from (26) we get

$$\tau^{-1} (\bar{z} - z) \in \arg\max_{u \in \mathcal{C}} \langle u, z - s \rangle, \quad \text{i.e.} \quad z'' \in \arg\max_{u \in \mathcal{C}} \langle u, -L^\top v - s \rangle\,. \tag{28}$$

Together, (27) and (28) show the saddle point property of (z'', v). Thus z'' must be a primal solution. □

By duality, the same scheme can be applied to solve problems where the *primal* constraint set is more complicated, i.e. $\mathcal{C} = \mathcal{C}_1 \cap \ldots \cap \mathcal{C}_r$. Also note that for $r = 1$, the algorithm reduces to the Douglas-Rachford method from [14]. In case both f and f_d can be numerically evaluated, the *gap* $f(z''^k) - f_d(v^k)$ provides a strong stopping criterion, as for any solution u^* and dual feasible point $v^k \in \mathcal{D}$,

$$f(z''^k) - f(u^*) \leqslant f(z''^k) - f_d(v^k)\,. \tag{29}$$

In practice, it is often better to stop depending on the *relative gap* $(f(u) - f_d(v))/f_d(v)$, which overestimates the actual gap and provides some scale invariance. However, in our case f usually cannot be evaluated due to the complexity of $\mathcal{D}$, and we must resort to a more elementary stopping criterion such as the difference between two consecutive iterates, $\|z''^k - z''^{k-1}\|$.

4 Experimental Results

We implemented and evaluated the proposed DMDR method as well as the fast primal-dual (FPD) [17] and Douglas-Rachford (DR) [14] methods in Matlab on an Intel Core2 Duo 2.66 GHz with 4 GB of RAM and 64-bit Matlab 2009a. The full data set for the experiments is available at `ipa.iwr.uni-heidelberg.de`.

Runtime Comparison. We compared the performance of the above algorithms on a four-class color segmentation problem (Fig. 3). The input image was generated by overlaying the synthetical "four colors" image with Gaussian noise, $\sigma = 1$. The data term was set to the ℓ_1-RGB distance to the four prototypical color vectors. For the regularizer we chose the Potts distance, $d(i,j) = \lambda$ iff $i \neq j$ and $d(i,j) = 0$ otherwise, with $\lambda = \sqrt{2}$. A reference solution and optimal dual objective f_d were computed using 5000 iterations of the DR method. The experiment was repeated 10 times with varying noise.

In terms of the number of iterations, the proposed DMDR method converges as fast as FPD. However, as it requires significantly less effort per iteration, it outperforms FPD and DR by a factor of $2-3$ with respect to total runtime.

High Label Count and Improved Numerical Robustness. For a larger number of labels, the runtime advantage is expected to become more apparent as the cost per iteration increases. We performed a 12-class segmentation of the

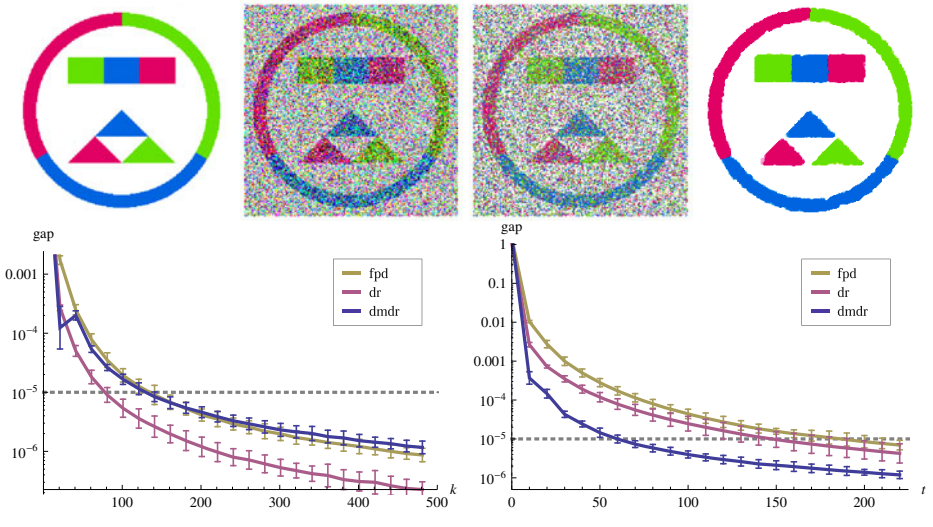


Fig. 3. Runtime comparison on a set of four-class labeling problems. **Top row, left to right:** Input image; input overlaid with heavy Gaussian noise; purely local labeling without regularizer; segmentation computed using the proposed method. The experiment was repeated 10 times with different noise. **Bottom row:** Gap vs. number of iterations (left) and time (right) with error indicators at 2σ. The proposed DMDR method performs comparable to FPD with respect to the number of iterations, but requires significantly less time per iteration, resulting in a total speedup of $2-3$.

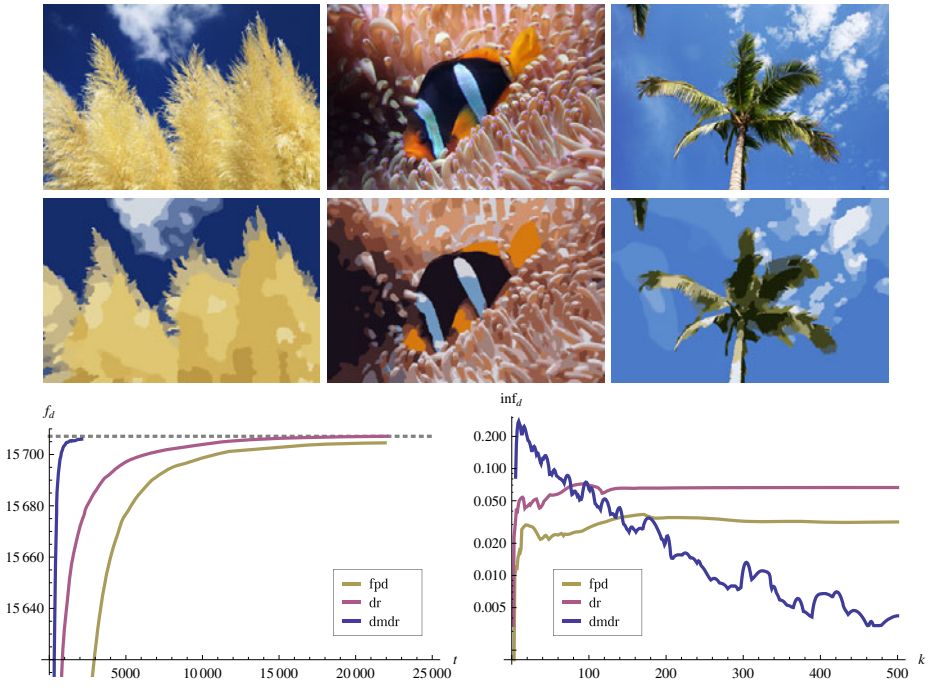


Fig. 4. Runtime performance on segmentation problems with a high label count. **Top row:** Input images (top) and segmentation into 12 classes (bottom) computed using the proposed DMDR method. **Bottom left:** Dual objective vs. time for 500 iterations on the "crop" image. The proposed method outperforms DR and FPD by a factor of 10 resp. 17. **Bottom right:** Infeasibility of the dual iterates vs. number of iterations. Due to the inexact projections, FPD and DR get stuck and converge to infeasible solutions. In contrast, DMDR gradually decreases the infeasibility to zero in theory and practice.

real-world images in Fig. 1 and Fig. 4 with the same data term as above with $\lambda = 0.2$ for the lake and fish images, and $\lambda = 0.5$ for the palm and crop images.

For this moderate number of labels, the iterative projections for DR and FPD are already quite slow, so we fixed a maximum of 5 inner iterations per outer step in order to get a reasonable computation time. The proposed method is about $6 - 10$ times faster than DR, and $7 - 17$ times faster than FPD (Fig. 4).

Moreover, due to the inexact projections, DR and FPD converge to infeasible dual points, i.e. they generate dual solutions v that do not lie inside the dual constraint set $\mathcal{D}$. In contrast, using DMDR the infeasibility gradually decreases, and is guaranteed to eventually drop to zero given exact arithmetic (Sect. 3).

Histogram-Based Segmentation and Absolute Distance. Fig. 5 shows the application of our method to a histogram-based three-class segmentation where the data term is based on probabilities computed from histograms over regions preselected by the user. In order to preserve more details, we chose $\lambda = 0.025$.

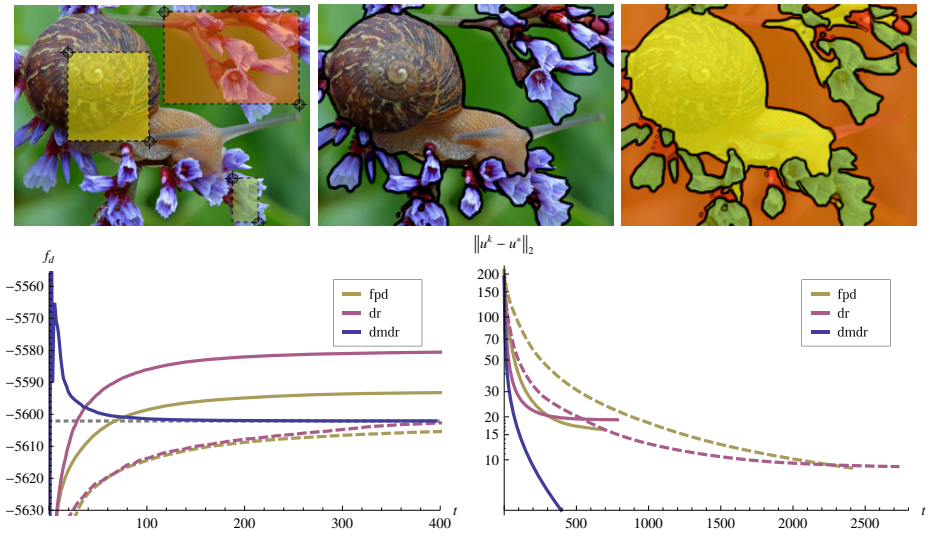


Fig. 5. Application to histogram-based segmentation. **Top row, left to right:** Input image with seed regions marked by the user; minimizer of the three-class variational segmentation using the proposed approach. **Bottom row:** Dual objectives (left) and ℓ_2 distance to the reference solution (right) vs. time. With low-accuracy approximate projections, FPD and DR get stuck in an infeasible solution (solid). Increasing the projection accuracy reduces the effect but slows down convergence (dashed). The proposed DMDR method avoids these problems and returns high-quality solutions after only a few iterations.

As above, it can be seen that FPD and DR get stuck at infeasible solutions, while DMDR converges smoothly. Increasing the accuracy of the approximate projections reduces the infeasibility, but leads to a much slower convergence.

It remains to ask how the dual gap relates to actual visual differences. Therefore at each step we evaluated the ℓ_2 distance of the current iterate to a reference solution computed using 5000 DMDR iterations (Fig. 5). Again it becomes clear that the inexact projections cause convergence issues for FPD and DR, while DMDR does not suffer from these problems. After 500 iterations, DMDR recovered a solution u^k with $\|u^k - u^*\|_2 \leqslant 10$, or $1.3 \cdot 10^{-4}$ per pixel, suggesting that only few iterations are required for visually high quality results.

Note that for all of the examples above, DMDR ran out of the box with $\tau = 1$, and did not require any parameter tuning.

Conclusion. We presented the DMDR method to efficiently solve saddle point problems with intricate dual constraints, as arise from tight relaxations of continuous multiclass labeling problems and general nonsmooth variational problems, using only simple operations that can easily be parallelized. Experiments indicate that it outperforms existing methods by a factor of $4 - 20$, and avoids the inaccuracies and convergence issues of the FPD and DR methods that rely on inexact projections.

References

1. Alberti, G., Bouchitté, G., Dal Maso, D.: The calibration method for the Mumford-Shah functional and free-discontinuity problems. Calculus of Variations and Partial Differential Equations 16(16), 299–333 (2003)
2. Ambrosio, L., Fusco, N., Pallara, D.: Functions of Bounded Variation and Free Discontinuity Problems. Clarendon Press (2000)
3. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. Patt. Anal. Mach. Intell. 23(11), 1222–1239 (2001)
4. Chambolle, A., Cremers, D., Pock, T.: A convex approach for computing minimal partitions. Tech. Rep. 649, Ecole Polytechnique CMAP (2008)
5. Chan, T.F., Esedoḡlu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. J. Appl. Math. 66(5), 1632–1648 (2006)
6. Douglas, J., Rachford, H.H.: On the numerical solution of heat conduction problems in two and three space variables. Trans. of the AMS 82(2), 421–439 (1956)
7. Eckstein, J.: Splitting Methods for Monotone Operators with Application to Parallel Optimization. Ph.D. thesis, MIT (1989)
8. Gabay, D.: Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems. In: Applications of the Method of Multipliers to Variational Inequalities, ch. IX, p. 299. North-Holland (1983)
9. Goldstein, T., Bresson, X., Osher, S.: Geometric applications of the split Bregman method: Segmentation and surface reconstruction. CAM Report 09-06, UCLA (2009)
10. Goldstein, T., Bresson, X., Osher, S.: Global minimization of Markov random fields with applications to optical flow. CAM Report 09-77, UCLA (2009)
11. Ishikawa, H.: Exact optimization for Markov random fields with convex priors. Patt. Anal. Mach. Intell. 25(10), 1333–1336 (2003)
12. Lellmann, J., Becker, F., Schnörr, C.: Convex optimization for multi-class image labeling with a novel family of total variation based regularizers. In: Int. Conf. Comp. Vis (2009)
13. Lellmann, J., Kappes, J., Yuan, J., Becker, F., Schnörr, C.: Convex multi-class image labeling by simplex-constrained Total Variation. In: Tai, X.-C., Mørken, K., Lysaker, M., Lie, K.-A. (eds.) SSVM 2009. LNCS, vol. 5567, pp. 150–162. Springer, Heidelberg (2009)
14. Lellmann, J., Schnörr, C.: Continuous multiclass labeling approaches and algorithms. Tech. rep., Univ. of Heidelberg (February 2010), http://www.ub.uni-heidelberg.de/archiv/10460/
15. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Comm. Pure Appl. Math. 42, 577–685 (1989)
16. Paragios, N., Chen, Y., Faugeras, O. (eds.): The Handbook of Mathematical Models in Computer Vision. Springer, Heidelberg (2006)
17. Pock, T., Cremers, D., Bischof, H., Chambolle, A.: An algorithm for minimizing the Mumford-Shah functional. In: Int. Conf. Comp. Vis. (2009)
18. Pock, T., Cremers, D., Bischof, H., Chambolle, A.: Global solutions of variational models with convex regularization. Tech. rep., Graz Univ. of Tech. (2009)
19. Rockafellar, R.: Convex Analysis. Princeton UP, Princeton (1970)
20. Setzer, S.: Split Bregman algorithm, Douglas-Rachford splitting and frame shrinkage. In: Tai, X.-C., Mørken, K., Lysaker, M., Lie, K.-A. (eds.) SSVM 2009. LNCS, vol. 5567, pp. 464–476. Springer, Heidelberg (2009)
21. Strang, G.: Maximal flow through a domain. Math. Prog. 26, 123–143 (1983)
22. Zach, C., Gallup, D., Frahm, J.M., Niethammer, M.: Fast global labeling for real-time stereo using multiple plane sweeps. In: Vis. Mod. Vis. (2008)

An Experimental Study of Color-Based Segmentation Algorithms Based on the Mean-Shift Concept

K. Bitsakos, C. Fermüller, and Y. Aloimonos

Center for Automation Research,
University of Maryland, College Park, USA
kbits@cs.umd.edu, {fer,yiannis}@cfar.umd.edu

Abstract. We point out a difference between the original mean-shift formulation of Fukunaga and Hostetler and the common variant in the computer vision community, namely whether the pairwise comparison is performed with the original or with the filtered image of the previous iteration. This leads to a new hybrid algorithm, called Color Mean Shift, that roughly speaking, treats color as Fukunaga's algorithm and spatial coordinates as Comaniciu's algorithm. We perform experiments to evaluate how different kernel functions and color spaces affect the final filtering and segmentation results, and the computational speed, using the Berkeley and Weizmann segmentation databases. We conclude that the new method gives better results than existing mean shift ones on four standard comparison measures ($\backsim$ 15%, 22% improvement on RAND and BDE measures respectively for color images), with slightly higher running times ($\backsim$ 10%). Overall, the new method produces segmentations comparable in quality to the ones obtained with current state of the art segmentation algorithms.

Keywords: image segmentation, image filtering, mean-shift.

1 Introduction

Mean shift is an unsupervised clustering technique that over the last decade gained popularity and is now widely used in computer vision for color based segmentation. Though conceptually simple, an extensive amount of mathematical formalism has been used to precisely describe the method. As a result, some of the important characteristics of the method were "hidden underneath the surface". This paper simplifies the formulation and brings forth its important features by describing mean shift as an optimization problem. This leads to two contributions; a) we propose a new variation, denoted Color Mean Shift, that combines Fukunaga's mean shift superior cluster ability with most of the computational advantages of Comaniciu's variant, and b) we experimentally compare different variations of the algorithm both in terms of the computational speed and the segmentation quality. Color Mean Shift is found to outperform the current methods in terms of the quality of segmentation, while it is slightly ($\backsim$ 10%)

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 506–519, 2010.

slower . More specifically, it produced $\backsim$ 15%, 22% better results on the Berkeley dataset with the RAND and the BDE measure respectively.

1.1 Related Work

Despite its existence for more than three decades [1], mean-shift only recently gained popularity in the computer vision community. Cheng [2] first modified the method and used it for non-parametric clustering and then, Comaniciu and Meer [3] used it for image filtering and segmentation. Since then, mean-shift has been used in computer vision for object tracking [4], 3D reconstruction [5], texture classification [6] and video segmentation [7] among other problems. The relatively high computational cost of a naive implementation of the method combined with the need for fast image processing led researchers to propose fast approximate variations of it. Most notably, two solutions for finding pairs of points within a radius have been proposed; the Improved Fast Gauss Transform based mean shift [8] for Normal kernels and the Locality Sensitive Hashing based mean shift [6].

Cheng [2] was the first to recognize the equivalence of mean shift to a step-varying gradient ascent optimization problem, and later Fashing and Tomashi [9] showed that it is equivalent to Newton's method with piecewise constant kernels, and is a quadratic bound maximization for all other kernels. Still the dominant way to describe it is by using density estimation terms [3], namely using kernels and their shadow and profile functions.

1.2 Contributions

In this paper, we describe mean shift as an optimization problem. The simplicity of the formulation not only leads to a better understanding of the method, but also brings forth the difference between the original method and its variation that is used in computer vision[1]. In the same section (Sec. 2), we propose our own variant of mean shift, denoted *Color Mean Shift* (CMS), that lies between the two methods. The next two sections contain an experimental comparison between the methods. First, in Sec. 3, we present the filtering results for different kernel functions and color spaces. Then, we study the filtering speed of the algorithms with respect to a number of optimization parameters. In Sec. 4 we show results on two different segmentation datasets (the Berkeley [10] and Weizmann Institute [11] databases) containing 300 images and 1387 human segmentations (in total) using 4 standard comparison measures. In these experiments the new method (i.e, color mean shift) exhibits an improvement of $> 15\%$ compared to the existing method on color images. A similar improvement was also achieved for the grayscale images of Weizmann dataset. Summary and future work (Sec. 5) conclude the paper.

[1] In the recent papers, the original "mean shift" approach is called "blurring mean shift". In the rest of the paper we use the abbreviations **FHMS** and **CMMS** for Fugunaga and Hostetler's and Comaniciu and Meer's method of mean shift, respectively.

2 Image Filtering Using the Mean Shift Algorithm

2.1 Notation

We consider the image on the 5D space with spatial and color dimensions. More specifically, $\mathbf{x}_i$ is a 2D vector representing the spatial coordinates and $\mathbf{s}_i$ is a vector that represents the three color channels of pixel i ($i = 1 \dots N$).

In the following paragraphs we use bold letters to represent vectors and the notation $[\mathbf{x}_i, \mathbf{s}_i]$ to indicate a concatenation of vectors. To indicate the evolution of a vector over time we use superscripts, eg. $[\mathbf{x}_i^0, \mathbf{s}_i^0]$ indicates pixel $\mathbf{x}_i$ having the initial intensity values $\mathbf{s}_i^0$.

2.2 Kernel Functions

In our experiments we use two different kernel functions; the Epanechnikov and the Normal (Gaussian) kernel. The Epanechnikov kernel has the analytic form

$$K_E(\mathbf{x}) = \begin{cases} c_E(1 - \mathbf{x}^T\mathbf{x}) & \mathbf{x}^T\mathbf{x} \leq 1 \\ 0 & otherwise \end{cases}, \tag{1}$$

where c_E is the normalization constant.

The multivariate Normal kernel with variance 1 has the analytic form

$$K_N(\mathbf{x}) = (2\pi)^{-\frac{d}{2}} \exp(-\frac{1}{2}\mathbf{x}^T\mathbf{x}). \tag{2}$$

The Normal kernel is symmetrically truncated to obtain a kernel with finite support.

2.3 Fukunaga and Hostetler's Mean Shift (FHMS)

The original mean shift formulation [1] (applied to a color image) treats the image as a set of $5 - D$ points. Each point is iteratively moved proportionally to the weighted average of its neighboring points. At the end, clusters of points are formed. We define mean shift to be the gradient descent solution of the optimization problem

$$\arg \min_{[\mathbf{x}_i, \mathbf{S}_i]} - \sum_{i,j} K([\mathbf{x}_i, \mathbf{s}_i] - [\mathbf{x}_j, \mathbf{s}_j]), \tag{3}$$

where $\sum_{i,j}$ defines the summation over all pairs of pixels in the image. This problem has a global minimum when all the pixels "collapse" into a single point. We seek a local minimum instead. That's why we initialize the features $[\mathbf{x}_i, \mathbf{s}_i]$ with the original position and color of the pixels of the image and perform gradient descent iterations till we reach the local minimum. The instabilities caused by this behavior are studied in a recent work of Rao et al. [12].

2.4 Comaniciu and Meer's Mean Shift (CMMS)

The modified mean shift formulation proposed by Comaniciu and Meer [3] (CMMS) can also be expressed as a gradient descent solution of the optimization problem

$$\arg \min_{[\mathbf{x}_i, \mathbf{S}_i]} - \sum_{i,j} K([\mathbf{x}_i, \mathbf{s}_i] - [\mathbf{x}_j^0, \mathbf{s}_j^0]). \tag{4}$$

There is a subtle difference between CMMS and FHMS, that significantly affects the behavior. In the former formulation each feature point is compared against the original set of $5 - D$ points $[\mathbf{x}_j^0, \mathbf{s}_j^0]$, while in the latter case the point is compared against the set of points from the previous iteration $[\mathbf{x}_j, \mathbf{s}_j]$.

Fig. 1 presents the results of both methods in a smoothly varying intensity image. Notice that the gradient of the kernel function, everywhere but in the boundaries, is zero and so CMMS filtering only changes the intensity on the boundaries (that change is not very visible). FHMS, on the other hand, produces artificial segments of uniform intensity. Intuitively, each iteration of the process results in more clustered data which in turn leads to better clustering results in the next iteration. On the downside, a fast FHMS implementation is challenging (if not impossible) due to the fact that the feature points and the comparison points do not lie on a regular spatial grid anymore. Thus one would have to compare the current feature $[\mathbf{x}_i, \mathbf{s}_i]$ against all the remaining feature points.

2.5 Color Mean Shift (CMS)

Our method *alleviates the computational problem* of FHMS by using the original spatial location of the points for comparison, while using the updated intensity values of the previous iteration for *improved clustering* ability. In a sense, we perform FHMS on the color dimensions and CMMS on the spatial dimensions (that is the reason for naming the method "color mean shift"). As above, CMS can be expressed as the gradient descent solution of the optimization problem

$$\arg \min_{[\mathbf{x}_i, \mathbf{S}_i]} - \sum_{i,j} K([\mathbf{x}_i, \mathbf{s}_i] - [\mathbf{x}_j^0, \mathbf{s}_j]). \tag{5}$$

We have included the results of color mean shift filtering in the smoothly varying image of Fig. 1. It is clear that individual clusters of uniform intensities are formed (as in the case of the original mean shift). Note that in this example there is not a single right solution for the segmentation problem and one can argue that a single segment is the best solution. We present this example only to exhibit one "weakness" of the CMMS algorithm, that is addressed in both our solution and the original mean shift algorithm. In Fig. 2 we present both CMS and CMMS algorithms.

(a) CMMS **(b)** CMS **(c)** FHMS

Fig. 1. All the described algorithms applied on a 256×100 pixels smoothly varying image. All the filtering algorithms were executed with spatial resolution $h_s = 21$ and range resolution $h_r = 10$ and used a Normal kernel.

CMS

Input:
set of pixels $\mathbf{x}_i^0$ with intensities $\mathbf{s}_i^0$
a function g
Output:
feature vector $[\mathbf{x}_i, \mathbf{s}_i]$
Algorithm:
initialize feature points $[\mathbf{x}_i, \mathbf{s}_i] \leftarrow [\mathbf{x}_i^0, \mathbf{s}_i^0]$
repeat until convergence
for all features $[\mathbf{x}_i, \mathbf{s}_i]$

$$[\mathbf{x}_i, \mathbf{s}_i] \leftarrow \frac{\sum_j [\mathbf{x}_j, \mathbf{s}_j] g(||[\mathbf{x}_i, \mathbf{s}_i] - [\mathbf{x}_j^0, \mathbf{s}_j]||^2)}{\sum_j g(||[\mathbf{x}_i, \mathbf{s}_i] - [\mathbf{x}_j^0, \mathbf{s}_j]||^2)}$$

CMMS

Input:
set of pixels $\mathbf{x}_i^0$ with intensities $\mathbf{s}_i^0$
a function g
Output:
feature vector $[\mathbf{x}_i, \mathbf{s}_i]$
Algorithm:
initialize feature points $[\mathbf{x}_i, \mathbf{s}_i] \leftarrow [\mathbf{x}_i^0, \mathbf{s}_i^0]$
for all features $[\mathbf{x}_i, \mathbf{s}_i]$
repeat until convergence

$$[\mathbf{x}_i, \mathbf{s}_i] \leftarrow \frac{\sum_j [\mathbf{x}_j, \mathbf{s}_j] g(||[\mathbf{x}_i, \mathbf{s}_i] - [\mathbf{x}_j^0, \mathbf{s}_j^0]||^2)}{\sum_j g(||[\mathbf{x}_i, \mathbf{s}_i] - [\mathbf{x}_j^0, \mathbf{s}_j^0]||^2)}$$

Connected Components Grouping

Input:
set of pixels $\mathbf{x}_i$ with intensities $\mathbf{s}_i$
grouping threshold t
Output:
label l_i for pixel $\mathbf{x}_i$
Algorithm:
repeat until convergence
for all pixels $\mathbf{x}_i$
for all $\mathbf{x}_j$ adjacent to $\mathbf{x}_i$
if $||s_i - s_j||^2 < t$ and x_i, x_j have different labels:
merge the labels of x_i and x_j $(l_i \equiv l_j)$

Fig. 2. In all algorithms $g(x) = [x \leq 1]$ (indicator function in Iverson notation) for the Epanechnikov and $g(x) = \exp(-x/2)$ for the Normal kernel

3 Filtering Comparison

Following the example of Comaniciu and Meer [3], we normalize the spatial and color coordinates of each pixel vector by dividing by the spatial (h_s) and color (h_r) resolutions. Thus, the original feature vector $[\mathbf{x}_i, \mathbf{s}_i]$ is transformed to $[\frac{\mathbf{x}_i}{h_s}, \frac{\mathbf{s}_i}{h_r}]$ (not included in the equations for simplicity). The spatial resolution

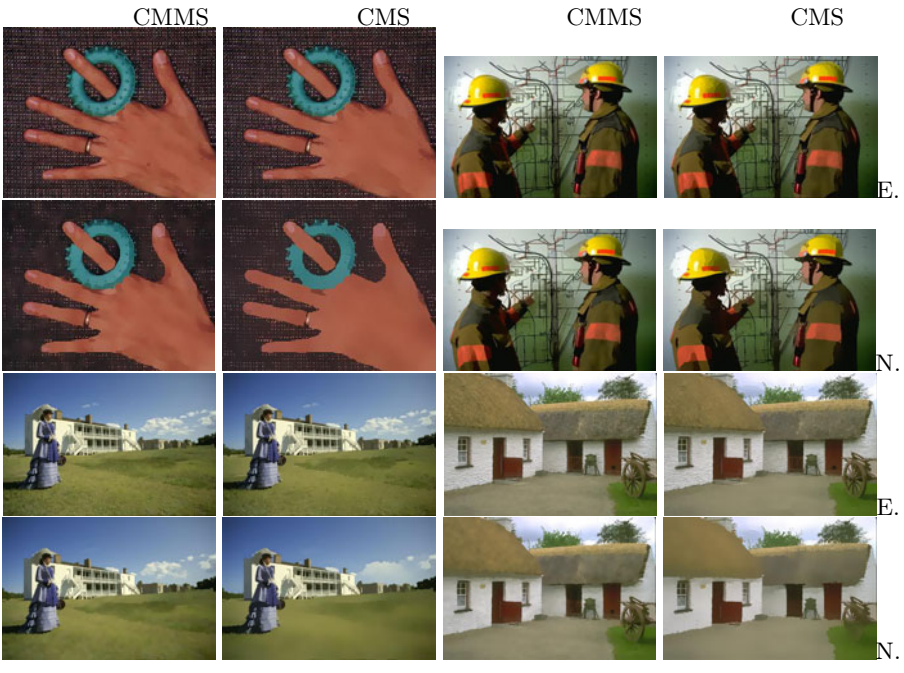


Fig. 3. Epanechnikov vs Normal kernel. We use $h_s = 5$ and $h_r = 19$. All images are processed in RGB color space. E., N. stand for Epanechnikov kernel and Normal kernel respectively. The Normal kernel produces smoother regions. Also, CMS produces more uniform regions even in heavily textured areas, eg. the grass and the roof.

h_s affects the size of the neighborhood around each pixel that the algorithm considers and in all the experiments is constant ($h_s = 5$ corresponding to a 11×11 window). Then, we perform the optimization; one pixel at a time in the case of CMMS (Fig. 2, top right), or one iteration of the whole feature set at a time for FHMS and CMS (Fig. 2, top left). FHMS has a complexity that is quadratic on the number of pixels of the whole image. Thus, its running time for a reasonably size image (eg. 640×480 pixels) is several minutes, making it prohibitively slow for any computer vision application. For that reason, we omit the results of this algorithm in the experiments.

3.1 Filtering Using an Epanechnikov or a Normal Kernel

First we present the effect of using different kernels: Epanechnikov and Normal (Fig. 3). Each column of the figure depicts the filtering result with a different algorithm (CMMS or CMS) and each row for a different kernel function (N., E. stand for Normal and Epanechnikov kernels respectively). In all cases the Normal kernel produces smoother results, while still preserving edge discontinuities. As a matter of fact, the color resolution h_r is the parameter that defines the gradient

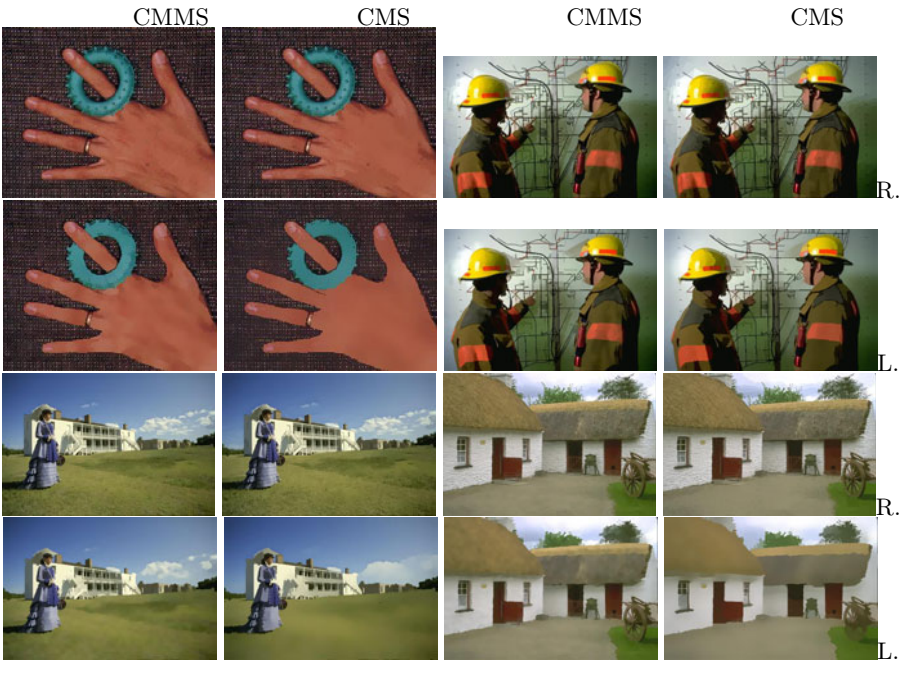


Fig. 4. Filtering in RGB vs Luv color space. We use $h_s = 5$ and $h_r = 5$. All images are processed with a Normal kernel. R, L stand for RGB and Luv respectively. Filtering in Luv makes smoother images. Moreover, CMS produces more uniform regions.

magnitude above which there is an edge (to be preserved). So for the "hand" image, a color range of $h_r = 19$ results in smoothing most of the texture of the background, while a value of $h_r = 10$ retains most of it (in RGB with a Normal kernel).

Overall CMS seems to produce more crisp boundaries between segments while creating more uniform regions within a segment (eg. it suppresses the skin color variation on the "hand" image). The former is particularly important for the segmentation step as we will see in Sec. 4.

3.2 RGB vs. Luv Color Space

In Fig. 4 we present the results when filtering on the RGB or Luv color space. In general, filtering in Luv produces smoother images.This is due to two facts; the Euclidean distance between two Luv values is perceptually meaningful, i.e., it is proportional to the distance of colors as perceived by a human observer, and the range of values for each component (L, u, v) is different (for example in our implementation $L \in [0 \dots 100]$, $u \in [-100 \dots 180]$, $v \in [-135 \dots 110]$.), while each of the red, green and blue components have values from 0 to 255.

Overall, CMS smoothes the image more than CMMS, while preserving the boundaries better.

3.3 Filtering Speed Comparison

With the increasing demand for processing large volumes of data computational speed has become an important characteristic of any algorithm, that along with accuracy determines its usefulness. That is the reason why a number of approaches to speed up mean shift filtering have been proposed [6,8]. In this section we try to compare the speed of the two methods.

An objective comparison of the filtering speed of the different methods is not a simple task. Besides the implementation details that greatly affect the speed, there is also a number of algorithmic parameters that can significantly speedup or slow down the convergence of the optimization procedure. We start our comparison by evaluating the role of these parameters and then we discuss whether general speed up techniques that have been proposed in the literature can be applied to the different methods or not. For fairness sake, we use our own implementation of all the filtering methods that consists of Matlab files for the image handling and the general input/output interface, while the optimization code is written in C[2]. We perform all the experiments on a desktop computer with an Intel Core2 Quad CPU @3GHz[3].

Image Size. In theory the complexity of both CMS and CMMS increases linearly with the number of pixels (if the kernel is bounded), since each pixel represents a feature vector that needs to be processed[4]. The theoretical prediction is verified in practice as Fig. 5a shows.

Spatial Resolution h_s. Theoretically, both filtering methods depend quadratically on the spatial bandwidth. In practice, other parameters, explained below, make the dependence less than quadratic. Fig. 5b displays the filtering speed with respect to the spatial resolution for the methods, when all the other parameters are the same.

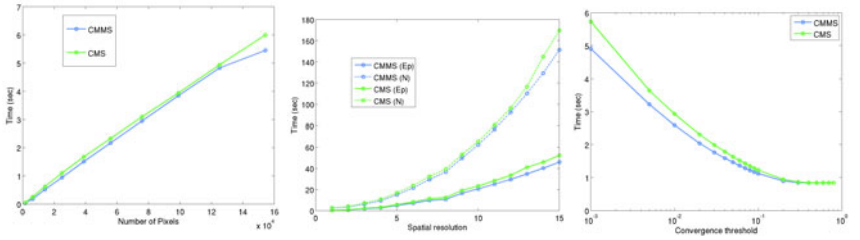
Epanechnikov vs. Normal kernel. For each pair of pixels, computation of the weight using the Epanechnikov kernel only requires a comparison, while the calculation of an exponential number is necessary for the case of the Normal kernel. As a result the former operation is much cheaper than the latter and thus filtering with an Epanechnikov kernel is faster compared to filtering with a Normal kernel as is shown in Fig. 5b.

The overall speed of the segmentation process is also affected by the quality of the result of the filtering process. We experimentally found, that a Normal kernel produced better results and as a consequence sped up the grouping step.

[2] All the code is available and can be downloaded from the author's website http://www.cs.umd.edu/~kbits/code.htm

[3] Due to Matlab's limitation only one core is used in the experiments.

[4] FHMS's complexity, on the other hand, is not linear with respect to the image size since whole areas can collapse into single points.



(a) Speed vs Image Size ($h_s = 5$)

(b) Speed vs h_s

(c) Speed vs Convergence threshold

Fig. 5. We use the "workers" image (size 321×481 pixels) and perform the filtering on the RGB color space with $h_r = 15$. A solid line denotes the use of Epanechnikov kernel while the dotted line (middle figure) the use of Normal kernel. We also limit the number of iterations to 20 and the convergence threshold is 0.001. We perform the filtering 5 times for each image size and only plot the median value.

The use of a Normal kernel still resulted in slower segmentation times, but the time difference was not as large as Fig. 5b shows.

Convergence Threshold. On each iteration of the optimization procedure each pixel vector is compared against its neighbors and shifted. If this shift is less than a predefined value (denoted convergence threshold) then we ignore that pixel in subsequent iterations of the optimization procedure. Intuitively the convergence threshold denotes how close to the "true" solution the optimization should reach before termination. Note that in CMMS the shift of each pixel is a monotonically decreasing function of the iteration number, while for CMS it is not. Fig. 5c displays the filtering speed with respect to the convergence threshold. The higher the threshold the faster the filtering. Especially for thresholds less than 0.1 the filtering time decreases almost exponentially.

Overall, from Fig. 5, CMS is $\backsim$ 10% slower than CMMS. A number of techniques can be used to perform the filtering faster. In the core of all filtering algorithms the pairwise distance between feature points needs to be computed. As suggested in [3] employing data structures and algorithms for multidimensional range searching can significantly improve the running time of all methods. In CMMS the trajectory of most feature points lay along the path of other feature points. Christoudias et al. [13] report a speed up of about five times when they "merge" the feature points together. This trick can directly be used in CMMS. A variation of the same concept could also be used to speed up CMS. The introduction of multicore CPUs and, especially, GPUs has provided a new way to improve the execution speed of algorithms through a parallel implementation. Both filtering algorithms are parallel in nature, so a careful implementation on a modern GPU is expected to run in real time for VGA or even larger sized images.

4 Segmentation Comparison

In a number of applications, like image denoising or deblurring, filtering is the final step. In most other applications filtering is an intermediate step followed by image segmentation. We are interested in the latter case. Thus, following the example of [3], we use the connected component grouping algorithm described in Fig. 2 to perform color-based segmentation. The simplicity of the grouping step allows for an objective evaluation of the filtering methods for the task of image segmentation. This algorithm has a single parameter, namely the grouping threshold t. In all our experiments $t = 0.5 * h_r$[5].

We use the Berkeley database of human segmentations [10] to evaluate the performance of the two methods. This is the biggest, publicly available database containing 200 color, training images and 1087 human created segmentations. We also present the results from the Weizmann Institute segmentation database [11], that consists of 100 grayscale images and 300 segmentations into foreground and background. Before presenting the results we need to describe the different measures that are used in the evaluation.

We use all the standard measures for the evaluation of the two algorithms, namely the Global Consistency Error (GCE) [10], the Variation of Information (VI) [14], the Probabilistic Rand index (PR) [15] and the average Boundary Displacement Error (BDE) [16].[6] From the previous measures for GCE, VI and BDE the lower the value the better the quality of the segmentation, while PR is a measure of similarity and as such a value of 0 indicates no similarity with the human created database, while a value of 1 indicates the highest similarity.

We create the following graphs by varying the color resolution h_r of the filtering methods. More specifically, we let h_r obtain values from 0.6 to 20 in increments of 0.3. We keep the remaining filtering parameters constant i.e., the maximum number of iterations for convergence is set to 20 and the convergence threshold to 0.1. For comparison we use the algorithm by Felzenswalb and Huttenlocher [18], denoted as GAT (Grouping with an Adaptive Threshold) on the figures. Again we vary the grouping threshold k ($k = [10 \dots 1500]$ in increments of 20).

We compute the comparison measures for each image of the database and further aggregate the results for the whole database using the median value[7]. These values are plotted on the Y-axis of each figure. On the X-axis we plot the average segment size, instead of the color resolution h_r. Thus all the plots below show the implicit curve of one comparison measure with respect to the average segment size.

[5] This is the same value for t that the EDISON system [13] uses. In practice, the threshold does not affect the resulting segmentation much, as long as it is larger than the convergence threshold of the optimization problem. In our experiments $t = 0.5 \gg 0.1 =$ convergence threshold.

[6] We use the code provided by J. Wright and A. Yang [17] to compute them.

[7] Since the comparison measures vary significantly for different images we choose the median value as opposed to the mean value because it is more robust to outliers.

4.1 Segmentation Results

First we present the collective segmentation results from the Berkeley database. We compare the two mean shift versions (CMMS and CMS) in two different color spaces (RGB and Luv) and using two different kernel functions (Epanechnikov and Normal kernel) for a total of $2 \times 2 \times 2 = 8$ combinations. That is why we display 8 curves on each graph of Fig. 6 plus a red curve for GAT.

Before analyzing the results any further we want to emphasize two facts. The results of the Global Consistency Error measure are misleading. As Martin et al. [10] mention, this measure only produces meaningful results when the number of segments in the computer segmentation is similar to the one in the human segmentation. In all other cases, i.e., when the number of computer generated segments is too high or too low GCE goes to zero. Indeed, as we observe in Fig. 6, all the curves for the GCE measure start from close to 0 (for very small average segment size) and asymptotically go to 0 (for very large average segment sizes). In between the two extremes, GCE values are larger, but since we display

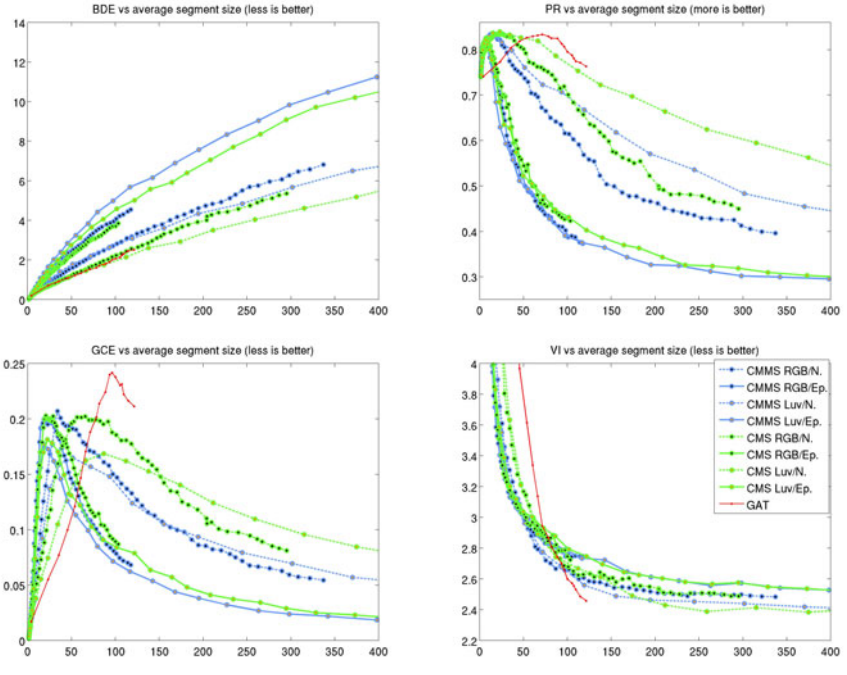


Fig. 6. Segmentation results for the Berkeley database. The solid and dash-dot lines represent the use of the Epanechnikov (Ep.) and Normal (N.) kernel, and the black and orange circle the use of the RGB and Luv color space, respectively. Note that the new method (CMS) is in *green*, while the existing method (CMMS) is in *blue*. From the top graphs it is clear that the green plots are better than the corresponding blue ones.

the average value for all the images it is impossible to determine the range of average segment sizes where GCE values are meaningful. The second fact is that the values of the Variation of Information measure for all the curves are really close together, making VI the least discriminative measure. On the other hand, both the Probabilistic Rand index and the average Boundary Displacement Error are discriminative enough to compare the different segmentation algorithms in this setting.

The segmentation results verify our earlier observations about the effect of the different kernels (Sec. 3.1) and color spaces (Sec. 3.2) on the amount of smoothing performed (for a given color resolution h_r). Filtering on the RGB color space results in less smoothing of the images and as a consequence in more image segments (and smaller average segment sizes). This is denoted by the close placement of the circles on the RGB plots compared to their Luv counterparts. The same observation, i.e., smaller average segment sizes, is valid for the Epanechnikov kernel function compared to the Normal kernel.

In the mean shift literature there are references that the Normal function produces better results than the Epanechnikov kernel [3], but so far a thorough analysis was not performed. According to the plots of Fig. 6 this prediction is absolutely right. The use of a Normal kernel produced better results in both measures (PR and BDE) and for both filtering methods (CMMS and CMS). Furthermore, the coupling of the Normal kernel with the Luv color space produced far superior results than all the other combinations.

Finally, the newly introduced variant of mean shift, i.e., Color Mean Shift, outperformed CMMS in all combinations of kernel functions and color spaces. Overall, CMS filtering on Luv color space with a Normal kernel produced the best results compared to all other methods. Compared to CMMS filtering on Luv color space with a Normal kernel (i.e., the next best algorithm) the new method produced on average $\backsim$ 17% better on the PR index and $\backsim$ 22% better on the BDE measure. Furthermore, this algorithm in most cases outperformed the current state of the art segmentation algorithm [18].

On Fig. 7 we present the segmentation results for the Weizmann dataset consisting of 100 images and 300 manual segmentations into foreground and background. Before analysing them we want to mention that this dataset is different from the previous one in the following aspects. All the images are grayscale and not color. Furthermore, the texture variation is significantly less than the one in the Berkeley database. The purpose of the dataset is to provide a testbed for segmentation into objects and as such only the single dominant object per image is marked as forground and the rest is background[8]. As a result many boundary edges are not reported in the manual segmentation. Both algorithms performed very well, with CMS performing better than CMMS on the BDE measure. In this database CMS performed slightly worse than GAT.

[8] The PR measure is misguiding in this dataset because of the existance of only two segments. Thus, a uniform segmentation of the whole image produces a result of ~ 0.97, i.e., very close to the maximum 1.

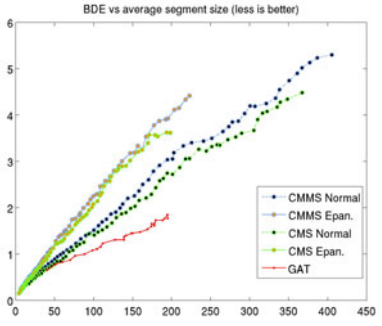


Fig. 7. Segmentation results for the Weizmann Institute database. The solid and dash-dot lines represent the use of the Epanechnikov (Epan.) and Normal kernel, respectively.

5 Conclusions

This paper presents the current variations of the mean shift algorithm from an optimization viewpoint and emphasizes the difference between Fukunaga's and Comaniciu's versions of the method, namely whether the pairwise comparison for moving each point is performed with the original image or with the filtered image of the previous iteration. A new variation of the mean shift algorithm, denoted Color Mean Shift, that lies between the existing two is also proposed. Extended experiments are presented both for the edge-preserving filtering and the segmentation tasks. In filtering, we mostly focus on the effect of different parameters on the speed of the filtering process. For segmentation, we use the Berkeley and the Weizmann Institute datasets to evaluate the performance of the algorithms using different kernel functions and color spaces. We conclude that Color Mean Shift performed on Luv color space using a Normal kernel function outperforms all other mean shift based algorithms for color images and is marginally better than current of the art segmentation algorithms. In the future we want to investigate how the methods perform when they are coupled with more sophisticated grouping techniques, such as [18].

Acknowledgements

The support of the EU under the Poeticon project (Cognitive Systems) is gratefully acknowledged.

References

1. Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function with applications in pattern recognition. IEEE Trans. Information Theory 21, 32–40 (1975)
2. Cheng, Y.: Mean shift, mode seeking, and clustering. PAMI 17, 790–799 (1995)

3. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Trans. on PAMI, 603–619 (2002)
4. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. PAMI 25, 564–577 (2003)
5. Wei, Y., Quan, L.: Region-based progressive stereo matching. In: CVPR, pp. 106–113 (2004)
6. Georgescu, B., Shimshoni, I., Meer, P.: Mean shift based clustering in high dimensions: A texture classification example. In: ICCV, pp. 456–463 (2003)
7. DeMenthon, D., Megret, R.: Spatio-temporal segmentation of video by hierarchical mean shift analysis. Technical report (2002)
8. Yang, C., Duraiswami, R., Gumerov, N., Davis, L.: Improved fast gauss transform and efficient kernel density estimation. In: ICCV, pp. 464–471 (2003)
9. Fashing, M., Tomasi, C.: Mean shift is a bound optimization. PAMI 27, 471–474 (2005)
10. 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)
11. Alpert, S., Galun, M., Basri, R., Brandt, A.: Image segmentation by probabilistic bottom-up aggregation and cue integration. In: CVPR, pp. 1–8 (2007)
12. Rao, S., Martins, A., Principe, J.: Mean shift: An information theoretic perspective. Pattern Recognition Letters 30, 222–230 (2009)
13. Christoudias, C., Georgescu, B., Meer, P.: Synergism in low-level vision. ICPR 4, 150–155 (2002)
14. Meila, M.: Comparing clusterings: an axiomatic view. In: ICML, pp. 577–584 (2005)
15. Unnikrishnan, R., Pantofaru, C., Hebert, M.: A measure for objective evaluation of image segmentation algorithms. In: Workshop on Empirical Evaluation Methods in Computer Vision, CVPR (2005)
16. Freixenet, J., Munoz, X., Raba, D., Marti, J., Cuff, X.: Yet another survey on image segmentation: Region and boundary information integration. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 408–422. Springer, Heidelberg (2002)
17. Yang, A.Y., Wright, J., Ma, Y., Sastry, S.: Unsupervised segmentation of natural images via lossy data compression. Comput. Vis. Image Underst. 110, 212–225 (2008)
18. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. IJCV 59, 167–181 (2004)

Towards More Efficient and Effective LP-Based Algorithms for MRF Optimization

Nikos Komodakis

University of Crete
Computer Science Department
komod@csd.uoc.gr

Abstract. This paper proposes a framework that provides significant speed-ups and also improves the effectiveness of general message passing algorithms based on dual LP relaxations. It is applicable to both pairwise and higher order MRFs, as well as to any type of dual relaxation. It relies on combining two ideas. The first one is inspired by algebraic multigrid approaches for linear systems, while the second one employs a novel decimation strategy that carefully fixes the labels for a growing subset of nodes during the course of a dual LP-based algorithm. Experimental results on a wide variety of vision problems demonstrate the great effectiveness of this framework.

1 Introduction

Message passing methods are extremely popular MRF optimization techniques in computer vision, with BP being the earliest method of this kind. Recently, many state of the art message-passing techniques have been proposed that rely on solving dual LP relaxations [1,2,3,4]. Compared to BP, they offer significant advantages such as better convergence properties, as well as the ability to provide suboptimality guarantees based on dual lower bounds. Moreover, they have been shown to significantly outperform BP and all other MAP estimation techniques [5]. On the other hand, one main drawback is that they often have a higher computational cost. As a result, given the large scale nature of the majority of vision problems, one of the key challenges in energy minimization is currently the acceleration of these methods. This is even more so considering the fact that computer vision researchers start gradually to resort to higher order MRF models, where such dual-based methods are expected to have much wider applicability due to their generality.

Motivated by the above observations, the goal of this work is to increase the overall efficiency of dual LP-based algorithms both for pairwise and higher order MRFs, while at the same time improving their effectiveness (*i.e.*, their accuracy). To this end, it proposes a framework that combines together two very general techniques in order to significantly speed up such algorithms. The first one is inspired by *algebraic multigrid* techniques for linear systems of equations, and uses a multiresolution hierarchy of dual relaxations for accelerating the convergence of dual-LP based methods. It relies on the premise that information

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 520–534, 2010.

is expected to propagate faster at lower resolutions. In the past, a geometric multigrid approach has been used for accelerating the BP algorithm for grid-structured graphs [6]. Here we extend and generalize such an approach to LP-based algorithms. Our algebraic multigrid framework can handle MRFs defined on any kind of graph, or having any kind of potentials. Moreover, it can be applied to higher order MRFs, as well as to LP relaxations that are tighter than the standard marginal polytope relaxation.

But to be able to achieve a significant speed up, besides accelerating the convergence, we also need to significantly reduce the time per iteration of a dual LP-based algorithm. To this end, we introduce a second technique, which consists of a decimation strategy that carefully fixes the labels for a growing subset of nodes during the course of the algorithm and thus one does not need to update their dual variables thereafter. It is based on the observation that, when using an algrebraic multigrid approach, a set of nodes typically exists that contribute a very small increase to the objective of the dual relaxation when their dual variables are updated. Similarly to the first technique, it is very general, and is applicable to both pairwise and higher order MRFs. Furthermore, it allows better primal solutions to be computed. Note that MRF decimation techniques have also been used in the past, and have been applied either to variants of BP [7,8] or to dual LP-based algorithms [9,10,11]. However, the latter techniques are not as widely applicable as our method.

After introducing in the next section the general setting used in the paper, we describe our framework in §3 - §7, while we discuss some extensions in §8. We present experimental results in §9 and finally conclude in §10.

2 Dual LP Relaxations for MRF Optimization

The problem of MAP estimation for discrete MRFs is typically formulated as follows. Given a graph $G = (\mathcal{V}, \mathcal{E})$ (where $\mathcal{V}$, $\mathcal{E}$ represent the nodes and edges of the graph) and a discrete set of labels $\mathcal{L}$, we want to assign a label x_p to each node p so that the total MRF energy (*i.e.*, the sum of all MRF potentials) is minimized, or

$$\mathrm{MRF}_G(\mathbf{U}, \mathbf{P}) := \min_{\mathbf{x}} \sum_{p \in \mathcal{V}} U_p(x_p) + \sum_{pq \in \mathcal{E}} P_{pq}(x_p, x_q) \ . \tag{1}$$

In the above, $\mathbf{U} = \{\mathbf{U}_p\}_{p \in \mathcal{V}}$ and $\mathbf{P} = \{\mathbf{P}_{pq}\}_{pq \in \mathcal{E}}$ denote respectively the set of all unary and pairwise potential functions.

As mentioned in the introduction, here we will concentrate on optimization methods that rely on dual LP relaxations. The most general setting for describing all these methods is based on the dual decomposition framework [3]. According to this framework, the original problem $\mathrm{MRF}_G(\mathbf{U}, \mathbf{P})$ (also called the master MRF) is decomposed into a set of simpler MRFs that are called the slaves and are denoted by $\mathrm{MRF}_{G_i}(\boldsymbol{\theta}^{G_i}, \mathbf{P})$. Here we assume that each slave MRF is defined on a subgraph $G_i = (\mathcal{V}_i, \mathcal{E}_i)$, has its own unary potentials (denoted by $\boldsymbol{\theta}^{G_i}$),

while it inherits[1] the pairwise potentials $\mathbf{P}$ of the master MRF. In this case, the dual variables are the unary potentials $\{\boldsymbol{\theta}^{G_i}\}$ of the slave MRFs, and the *key property* that these variables have to satisfy is $\sum_i \boldsymbol{\theta}^{G_i} = \mathbf{U}$, *i.e.*, the sum of the unary potentials of the slaves should give back the unary potentials of the master MRF.

Based on this property, it is easy to prove that the sum of the optimal energies of the slaves always provides a lower bound to the optimal energy of the master, and so the goal of the dual LP relaxation is exactly to adjust the dual variables so as to maximize this lower bound, or

$$\max_{\{\boldsymbol{\theta}^{G_i}\}} \sum_i \mathrm{MRF}_{G_i}(\boldsymbol{\theta}^{G_i}, \mathbf{P}) \tag{2}$$

$$\text{s.t.} \sum_i \boldsymbol{\theta}^{G_i} = \mathbf{U} \ . \tag{3}$$

Different dual-based optimization algorithms have been proposed in the literature, all of which try to solve the above dual relaxation, and the key property that has to be maintained (either implicitly or explicitly) is condition (3).

3 Accelerating Dual LP-Based Optimization Algorithms via an Algrebraic Multigrid Approach

Due to the decomposition of the master MRF into a set of smaller slave MRFs, the update of the dual variables is essentially done based only on local information. As a result, information travels slowly across the graph, and this has the undesirable effect of slowing down the convergence of dual LP-based algorithms, which thus require many iterations to converge to the correct solution. This issue is essentially very similar to the slow convergence problem faced by iterative algorithms for linear systems. Again, due to the local nature of the updates, such algorithms can recover very fast (*i.e.*, in few iterations) the high-frequency part of the solution, but they are very slow at recovering the lower frequencies. Multigrid is introduced to overcome this problem, where the basic idea is based on the trivial observation that low frequencies in the original grid reappear as high frequencies in a grid of lower resolution. A multigrid approach thus replaces the original linear system with a hierarchical multiresolution set of linear systems. The two key elements in a multigrid algorithm are the so called restriction and prolongation operators, that specify the transition between linear systems at adjacent levels in the hierarchy. These operators are combined to generate a so called V-cycle, which consists of a fine-to-coarse restriction phase followed by a coarse-to-fine prolongation phase.

Our aim here will be to apply a similar strategy to dual based MRF algorithms for quickly solving (2). This will be done by using a hierarchy of dual decompositions, defined on a sequence of graphs $G = G^{(0)}, G^{(1)}, \ldots, G^{(T)}$, where each

[1] In general, each slave can have its own pairwise potentials (just like the unary potentials) and does not need to inherit them from the master MRF. Here we assume they are inherited only to simplify the presentation and to reduce notational clutter, but everything described can be very easily extended to the more general case.

graph $G^{(t+1)}$ is assumed to be a "coarser" version of graph $G^{(t)}$ (we will explain what precisely we mean by "coarser" later). We will also define a restriction and prolongation operator, denoted hereafter by PROJ and LIFT respectively. The role of the restriction operator will be to take as input a master MRF and its dual decomposition at level t, and to project them onto level $t+1$, *i.e.*, create a corresponding master problem and a corresponding dual decomposition at level $t+1$

$$\begin{array}{c} \mathrm{MRF}_{G^{(t)}}(\mathbf{U}^{(t)}, \mathbf{P}^{(t)}) \\ \left\{\mathrm{MRF}_{G_i^{(t)}}(\boldsymbol{\theta}^{G_i^{(t)}}, \mathbf{P}^{(t)})\right\} \end{array} \xrightarrow{\text{PROJ}} \begin{array}{c} \mathrm{MRF}_{G^{(t+1)}}(\mathbf{U}^{(t+1)}, \mathbf{P}^{(t+1)}) \\ \left\{\mathrm{MRF}_{G_i^{(t+1)}}(\boldsymbol{\theta}^{G_i^{(t+1)}}, \mathbf{P}^{(t+1)})\right\} \end{array} \tag{4}$$

On the contrary, the role of the prolongation operator LIFT will be to take as input a feasible set of dual variables $\left\{\boldsymbol{\theta}^{G_i^{(t+1)}}\right\}$ for the decomposition defined at the "coarser" level $t+1$, and to lift them to a feasible set of dual variables $\left\{\boldsymbol{\theta}^{G_i^{(t)}}\right\}$ for the decomposition that has been previously defined at level t, *i.e.*,

$$\left\{\boldsymbol{\theta}^{G_i^{(t+1)}}\right\} \xrightarrow{\text{LIFT}} \left\{\boldsymbol{\theta}^{G_i^{(t)}}\right\} \ . \tag{5}$$

Just like in multigrid, a V-cycle in our case will consist of a restriction phase followed by a prolongation phase (see Fig. 1(a)). In the restriction phase we sequentially apply operator PROJ to all but the last level in the hierarchy, *i.e.*, we start from level $t = 0$ and go up to level $t = T - 1$. In this manner, a master MRF along with a dual decomposition is generated for each level. All of these decompositions are essentially projections of the original master problem and its dual decomposition. In the prolongation phase, we move in the opposite direction. This means that for each level t (where t now starts from $t = T$ and terminates at $t = 0$) we solve the dual relaxation corresponding to the decomposition at that level, and then we lift the resulting solution onto the next finer level (if one exists) via using the operator LIFT, thus initializing the dual variables for the decomposition at level $t - 1$. Due to the the information traveling much faster at the "coarser" levels of the hierarchy, the dual relaxations for these levels can be solved very fast, *i.e.*, in very few iterations. Furthermore, this quick spreading of the information that took place in the coarser levels is carried over to the finer levels, thanks to the initialization of the dual variables via the LIFT operator (assuming, of course, that this operator has been properly defined, which is crucial for the success of this scheme). This, in turn, results into accelerating the convergence of the dual relaxations at the finer levels as well.

Having explained the overall structure of our method, it still remains to describe how to generate the hierarchy of graphs, how the master problems and their dual decompositions are defined at each level, and, most importantly, how to efficiently compute the operators LIFT and PROJ, which is, of course, one of the key technical issues. Before doing so, we must note that we want our scheme to be applicable to any kind of graph G, and not only to grids, as well as to MRFs with any kind of potential functions. Drawing an analogy with multigrid methods, we want to derive an algebraic (and not a geometric) multigrid solver, as the former is much more widely applicable.

4 Defining the Hierarchy of Graphs

Given a graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{p_1, p_2, \ldots, p_n\}$, we want to define a "coarser" graph $\bar{G} = (\bar{\mathcal{V}}, \bar{\mathcal{E}})$. All that is needed as input for this purpose, is a partition $\{\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_{\bar{n}}\}$ of $\mathcal{V}$, *i.e.*, $\cup \bar{p}_i = \mathcal{V}$ and $\bar{p}_i \cap \bar{p}_j = \emptyset$. Each node of the "coarser" graph $\bar{G}$ will then correspond to a subset of this partition, *i.e.*, it will hold $\bar{\mathcal{V}} = \{\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_{\bar{n}}\}$, where we hereafter use $\bar{p}_i$ to denote both a node of $\bar{\mathcal{V}}$ as well as a subset of nodes of $\mathcal{V}$. Under this convention, the projection (denoted by $\text{proj}(p)$) of a node $p \in \mathcal{V}$ is defined as the unique node $\bar{p} \in \bar{\mathcal{V}}$ that satisfies the condition $p \in \bar{p}$, while the projection of a subset of nodes $\{p_k\} \subseteq \mathcal{V}$ is naturally equal to the union of the individual projections, *i.e.*, $\text{proj}(\{p_k\}) = \cup \text{proj}(p_k)$. Based on this notation, the set of edges $\bar{\mathcal{E}}$ of $\bar{G}$ is then defined as $\bar{\mathcal{E}} = \{\text{proj}(p_i p_j) | p_i p_j \in \mathcal{E}, \text{proj}(p_i) \neq \text{proj}(p_j)\}$. The resulting "coarser" graph $\bar{G} = (\bar{\mathcal{V}}, \bar{\mathcal{E}})$ is called the projection of graph G, and is denoted by $\text{proj}(G)$ (*e.g.*, see Fig. 1(b)). Therefore, to define a hierarchy of graphs, it suffices to set $G^{(t+1)} = \text{proj}(G^{(t)})$, where we assume that a partition has been specified by the user for each of the projections and $G^{(0)} = G$.

Assigning a label to a node $\bar{p} \in \bar{\mathcal{V}}$ of the "coarser" graph $\bar{G} = \text{proj}(G)$ will mean that this label is assigned to all nodes of G in the set $\{p \in \mathcal{V} \,|\, \text{proj}(p) = \bar{p}\}$. Based on this convention, if $\text{MRF}_G(\mathbf{U}, \mathbf{P})$ is an MRF[2] on the graph G, its projection on $\bar{G}$ will be an MRF, denoted by $\text{proj}(\text{MRF}_G(\mathbf{U}, \mathbf{P})) := \text{MRF}_{\bar{G}}(\bar{\mathbf{U}}, \bar{\mathbf{P}})$, whose potentials $\bar{\mathbf{U}}$, $\bar{\mathbf{P}}$ are defined as follows[3]:

$$\bar{U}_{\bar{p}}(l) = \sum\nolimits_{p:\text{proj}(p)=\bar{p}} U_p(l), \quad \bar{P}_{\bar{p}\bar{q}}(l, l') = \sum\nolimits_{pq:\text{proj}(pq)=\bar{p}\bar{q}} P_{pq}(l, l') \ . \tag{6}$$

Naturally, we want the master MRF at each level of our hierarchy to be a projection of the original MRF.

5 Defining the Restriction Operator PROJ

It suffices to show how to define this operator for one level of the hierarchy, *i.e.*, during a transition from a graph G to a coarser graph $\bar{G} = \text{proj}(G)$. Let $\text{MRF}_G(\mathbf{U}, \mathbf{P})$ be the master MRF on G, and let $\left\{\text{MRF}_{G_i}(\boldsymbol{\theta}^{G_i}, \mathbf{P})\right\}$ be its dual decomposition (*i.e.*, the set of slaves defined on subgraphs $\{G_i\}$). The main role of operator PROJ will be to define the corresponding dual decomposition for the graph $\bar{G}$, denoted by $\left\{\text{MRF}_{\bar{G}_j}(\bar{\boldsymbol{\theta}}^{\bar{G}_j}, \bar{\mathbf{P}})\right\}$. To this end, it first needs to determine the set of subgraphs $\{\bar{G}_j\}$ on which the new slaves will be defined. This set will consist of all subgraphs of the form $\text{proj}(G_i)$, *i.e.*,

$$\{\bar{G}_1, \bar{G}_2, \ldots, \bar{G}_{\mathcal{J}}\} = \{\text{proj}(G_1), \text{proj}(G_2), \ldots, \text{proj}(G_{\mathcal{I}})\} \ . \tag{7}$$

[2] Depending on the context, $\text{MRF}_G(\mathbf{U}, \mathbf{P})$ denotes either a MRF (on a graph G) with unary and pairwise potentials $\mathbf{U}$, $\mathbf{P}$ or a minimum MRF energy (as in (1)).

[3] To reduce notational clutter, we assume it holds $P(l, l) = 0$ when defining the potentials $\bar{\mathbf{U}}$, otherwise we must set $\bar{U}_{\bar{p}}(l) = \sum_{p:\text{proj}(p)=\bar{p}} U_p(l) + \sum_{pq:\text{proj}(pq)=\bar{p}} P_{pq}(l, l)$.

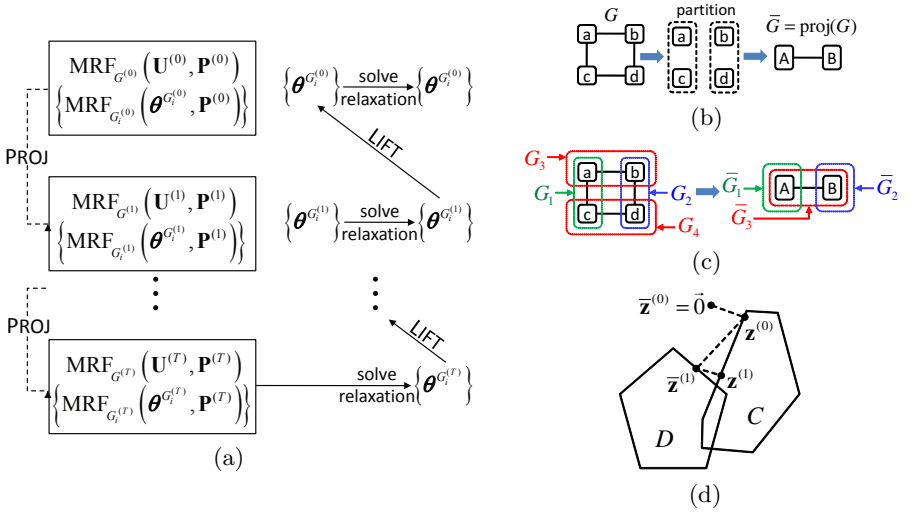


Fig. 1. (a) V-cycle of the 'algrebraic multigrid' approach for dual LP-based algorithms **(b)** $\bar{G}$ is the projection of graph G based on the partition $\{a, c\}, \{b, d\}$. **(c)** If G_1, G_2, G_3, G_4 are the subgraphs of the slaves in G, then $\bar{G}_1$, $\bar{G}_2$, $\bar{G}_3$ will be the subgraphs of the 3 slaves in $\bar{G}$. Note that $\bar{G}$ has fewer slaves since both G_3, G_4 project onto $\bar{G}_3$. Also note that the slaves for $\bar{G}_1$, $\bar{G}_2$ have no pairwise potentials. **(d)** The projection of $\mathbf{0}$ onto $C \cap D$ is computed via alternating projections on C and D (note that although C and D are drawn here as polytopes, they are actually affine subspaces in our case).

Since it can hold $\text{proj}(G_i) = \text{proj}(G_{i'})$ for $i \neq i'$, it is important to emphasize that the number of subgraphs $\bar{G}_j$ may be *strictly less* than the number of subgraphs G_i (see Fig. 1(c)). The operator PROJ then associates to each different subgraph $\bar{G}_j$ a slave $\text{MRF}_{\bar{G}_j}(\bar{\boldsymbol{\theta}}^{\bar{G}_j}, \bar{\mathbf{P}})$ whose potential functions are defined as follows:

$$\bar{\boldsymbol{\theta}}_{\bar{p}}^{\bar{G}_j}(l) = \sum\nolimits_{i:\text{proj}(G_i)=\bar{G}_j} \sum\nolimits_{p:\text{proj}(p)=\bar{p}} \boldsymbol{\theta}_p^{G_i}(l) \ , \tag{8}$$

$$\bar{P}_{\bar{p}\bar{q}}(l, l') = \sum\nolimits_{pq:\text{proj}(pq)=\bar{p}\bar{q}} P_{pq}(l, l') \ , \tag{9}$$

i.e., essentially it holds $\text{MRF}_{\bar{G}_j}(\cdot, \cdot) = \sum_{i:\text{proj}(G_i)=\bar{G}_j} \text{MRF}_{G_i}(\cdot, \cdot)$. Eqs. (7)-(9) completely specify the dual decomposition for graph $\bar{G}$. Furthermore, this decomposition, in turn, completely specifies the potentials of the master MRF for $\bar{G}$, denoted by $\text{MRF}_{\bar{G}}(\bar{\mathbf{U}}, \bar{\mathbf{P}})$, since it must hold $\bar{\mathbf{U}} = \sum_j \bar{\boldsymbol{\theta}}^{\bar{G}_j}$ due to (3). However, there still remains one critical question that must be answered: is the resulting master MRF a projection onto $\bar{G}$ of the master MRF for G, as we want? It turns out that this is indeed the case, as the following theorem certifies:

Theorem 1 ([12]). *If* $\text{MRF}_{\bar{G}}(\bar{\mathbf{U}}, \bar{\mathbf{P}})$ *is the master MRF resulting from the dual decomposition defined by eqs.* (7)-(9)*, it then holds* $\text{MRF}_{\bar{G}}(\bar{\mathbf{U}}, \bar{\mathbf{P}}) = \text{proj}(\text{MRF}_G(\mathbf{U}, \mathbf{P}))$.

6 Defining the Prolongation Operator LIFT

Let $\text{MRF}_G(\mathbf{U}, \mathbf{P})$, $\{\text{MRF}_{G_i}(\boldsymbol{\theta}^{G_i}, \mathbf{P})\}$ and $\text{MRF}_{\bar{G}}(\bar{\mathbf{U}}, \bar{\mathbf{P}})$, $\{\text{MRF}_{\bar{G}_j}(\bar{\boldsymbol{\theta}}^{\bar{G}_j}, \bar{\mathbf{P}})\}$ be the master MRFs along with their set of slaves for two graphs G, $\bar{G} = \text{proj}(G)$ that are adjacent in the hierarchy. We assume that all these MRFs have been constructed during the restriction phase. We are now at the prolongation phase, where we assume that the dual relaxation for $\bar{G}$ has already been solved (*i.e.*, the dual variables $\{\bar{\boldsymbol{\theta}}^{\bar{G}_j}\}$ are set to their optimal values), and we now want to compute the LIFT operator whose role is to initialize the dual variables $\{\boldsymbol{\theta}^{G_i}\}$ for graph G. Note that, since $\{\bar{\boldsymbol{\theta}}^{\bar{G}_j}\}$ are already set to their optimal values, this implies that an important amount of information has already been spread across the whole graph $\bar{G}$ (and hence across G as well, since $\bar{G} = \text{proj}(G)$). Therefore, if we manage to properly take into account this information when initializing $\{\boldsymbol{\theta}^{G_i}\}$, we will succeed in accelerating the convergence of the dual relaxation for graph G as well.

But how can we go about doing that? A first idea that comes in mind is the following one: Let $\text{Opt}_{\bar{G}}$ be the already computed optimal value of the dual relaxation for $\bar{G}$. Recall that our goal is to maximize the dual objective function for graph G as well. Therefore, perhaps we should aim at initializing the dual variables $\{\boldsymbol{\theta}^{G_i}\}$ such that the resulting dual objective is at least as large as $\text{Opt}_{\bar{G}}$. Unfortunately, this is not, in general, possible, as the following theorem shows:

Theorem 2 ([12]). *Let* $\text{Opt}_{\bar{G}}$, Opt_G *denote the optimal values of the dual relaxations for graphs* $\bar{G}$ *and* G *respectively. Then, in general, it holds* $\text{Opt}_{\bar{G}} > \text{Opt}_G$.

However, dual variables $\{\bar{\boldsymbol{\theta}}^{\bar{G}_j}\}$ still provide very important information about dual variables $\{\boldsymbol{\theta}^{G_i}\}$ that we can take advantage of. In particular, they provide the linear constraints (8), where values $\bar{\boldsymbol{\theta}}_{\bar{p}}^{\bar{G}_j}(\cdot)$ are now assumed to be known. By imposing these constraints when initializing variables $\{\boldsymbol{\theta}^{G_i}\}$, we implicitly take into account all information that is encoded in $\{\bar{\boldsymbol{\theta}}^{\bar{G}_j}\}$ and has propagated across graph $\bar{G}$. Of course, besides eqs. (8), $\{\boldsymbol{\theta}^{G_i}\}$ must also satisfy the dual feasibility constraints (3). Therefore, in total, variables $\{\boldsymbol{\theta}^{G_i}\}$ should be initialized so as to satisfy the linear system composed of Eqs. (3) and (8). Among the many solutions of this underdetermined linear system, we must compute the one that has minimum Euclidean norm. Intuitively, this regularization of the solution is important because otherwise the resulting initial dual variables $\{\boldsymbol{\theta}^{G_i}\}$ for the finer graph may exhibit large variations in magnitude, which can have as a result that too much energy/information is concentrated on local parts of the fine graph. This can destroy the propagation of "information" that took place at the coarser level and can thus hinder convergence. We next show how to efficiently perform this minimum norm computation.

6.1 Solving for $\{\boldsymbol{\theta}^{G_i}\}$

During this section, in order to make the exposition more clear, we will use $\mathbf{z} = \{z_k\}_{k=1}^K$ to denote the vector from concatenating all $\{\boldsymbol{\theta}^{G_i}\}$. Our goal is to find the least norm solution of an underdetermined linear system, *i.e.*,

$$\min_{\mathbf{z}} \|\mathbf{z}\|^2 \tag{10}$$

$$\text{s.t. } \mathbf{A}\mathbf{z} = \mathbf{b} \ , \tag{11}$$

where $\mathbf{A}\mathbf{z} = \mathbf{b}$ encodes the linear constraints (3) and (8). Theoretically, such a $\mathbf{z}$ can be computed as $\mathbf{z} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{b}$, but this may be too slow for our purposes. Fortunately, a solution to (10) can be computed extremely fast by exploiting the special structure existing in the constraints (3), (8). To this end, we first rewrite the above optimization problem as follows:

$$\min_{\mathbf{z}} \|\mathbf{z} - \mathbf{0}\|^2 \tag{12}$$

$$\text{s.t. } \mathbf{z} \in C \cap D \ , \tag{13}$$

where C, D denote the linear subspaces of $\mathbb{R}^K$ corresponding to the linear equations (3) and (8) respectively. Therefore, the optimal $\mathbf{z}$ coincides with the orthogonal projection of the zero vector onto the intersection of the two linear subspaces C and D. To compute this projection, we apply the well known Dykstra algorithm [13], which is an alternating projection method, *i.e.*, it starts from the zero vector $\bar{\mathbf{z}}^{(0)} = \mathbf{0}$, and then alternately projects onto C and D:

$$\mathbf{z}^{(n)} = \mathcal{P}_C(\bar{\mathbf{z}}^{(n)}), \quad \bar{\mathbf{z}}^{(n+1)} = \mathcal{P}_D(\mathbf{z}^{(n)}), \quad n = 0, 1, 2, \ldots \tag{14}$$

where $\mathcal{P}_C(\cdot)$ and $\mathcal{P}_D(\cdot)$ denote projection onto C and D, respectively (see Fig 1(d)). This generates a sequence $\mathbf{z}^{(n)} \in C$ which provably converges to the optimal solution. The advantage in doing so is that the projections $\mathcal{P}_C(\cdot), \mathcal{P}_D(\cdot)$ can be computed extremely fast in our case due to the special structure of the linear subspaces C and D. Namely, it is easy to verify that both subspaces are specified by a set of equations of the following form:

$$\sum_{k \in I_j} z_k = b_j, \quad j = 1, 2, \ldots, J \ , \tag{15}$$

where the sets $\{I_j\}_{j=1}^J$ form a partition of the set of indices $I = \{1, 2, \ldots, K\}$, *i.e.*, $\cup_j I_j = I$ and $I_j \cap I_{j'} = \emptyset$ for $j \neq j'$. The projection of a point $\mathbf{z}'$ onto such a linear subspace is easily seen to be given by the following vector $\mathbf{z}$:

$$\forall k \in I_j, \quad z_k = z'_k + (b_j - \sum_{i \in I_j} z'_i)/|I_j|, \quad j = 1, 2, \ldots, J \ . \tag{16}$$

Furthermore, the Dykstra algorithm converges very fast in our case (*i.e.*, extremely few alternating projections are required). Theoretically this can be attributed to the fact that the rate of convergence of this algorithm increases with the angle $\theta \in [0, \frac{\pi}{2}]$ between the two subspaces, *i.e.*, the more orthogonal the subspaces are, the faster the convergence. Hence, overall, this algorithm leads to a very fast method for minimizing (10), *i.e.*, for initializing $\{\boldsymbol{\theta}^{G_i}\}$.

7 Accelerating Dual LP-Based Methods via Fixing Variables

The multigrid approach described above allows information to propagate faster across the MRF graph, and this helps us to reduce the number of iterations to convergence at the finest level. But to be able to take full advantage of this fact and achieve a significant speed up, we also need to reduce the time spent per iteration at that level. To this end, we now describe a technique that is applied only at the finest level of the hierarchy during the multigrid approach. As mentioned above, its main role is to bring a significant reduction in the time per iteration at that level (but, in addition to that, it also helps us to speed up the convergence of the algorithm). This reduction is achieved via a decimation strategy, where we carefully fix the labels for a dynamically growing subset of nodes during the algorithm, and do not update their dual variables thereafter. Recall that the cost of an iteration essentially comes from locally updating the dual variables $\{\boldsymbol{\theta}_p^{G_i}(\cdot)\}$ for each node p in the graph. These updates aim to improve the dual objective. However, it is often the case that the rate of improvement per iteration is very small despite the great computational effort, *i.e.*, the dual function increases only slightly per iteration, and this in turn leads to a slow progress towards a good primal solution. The reason for this behaviour comes from the fact that many nodes cannot contribute a positive increase when their local dual variables are updated during an iteration. The following definition is important in this regard: we say that a node p is *stabilized* at the t-th iteration if, exactly before the update of the local variables $\{\boldsymbol{\theta}_p^{G_i}(\cdot)\}$ at that iteration, there exists a label that optimizes all the current instances of slaves containing p (any such label will be called *stable* w.r.t. p). It is easy to verify the following proposition:

Proposition 1 ([12]). *If a node p is stabilized then no update of its local dual variables $\{\boldsymbol{\theta}_p^{G_i}(\cdot)\}$ can increase the dual objective. Conversely, if p is non-stabilized, then there always exists an update of variables $\{\boldsymbol{\theta}_p^{G_i}(\cdot)\}$ that improves the dual.*

According to this proposition, for example, stabilized nodes leave the dual function unmodified in sequential algorithms such as TRW-S or max-diffusion. But stabilized nodes also lead us to the central concept in our decimation method, that of an *R-nested* node: we say that node p is *R-nested* for the t-th iteration if both p and all other nodes of graph G within distance[4] R from p were found to be stabilized at that iteration (see Fig. 2(a)). Motivated by proposition 1, we have empirically verified the following two important observations: in practice, many nodes quickly become stabilized during a dual-based algorithm when a multigrid scheme is used, and, furthermore, stabilized nodes that consistently remain R-nested for a number of iterations (with R large enough) turn out to contribute a very small (even zero) total change to the dual objective thereafter. This leads to the following decimation strategy (that depends on two positive

[4] The distance of two nodes is the number of edges of their minimum connecting path.

integer parameters R, D): at each iteration, we fix all nodes that are stabilized at the current iteration and that were R-nested for the past D iterations (each such node is simply assigned one of its current stable labels). This strategy is applied after a few initial iterations have passed, while parameters R and D determine how fast nodes can become fixed, and must be set to some reasonably large values.

To intuitively understand the necessity for the conditions of the above decimation strategy notice that an R-nested node is essentially surrounded by a 'layer' (of width R) of stabilized nodes. Note also that if a node, say q, becomes non-stabilized at the current iteration, this means that q is able to contribute to the dual objective. This in turn implies that extra dual information (in the form of messages) can originate from q and propagate to nearby nodes, thus possibly affecting the labels of any node p within a certain distance, say R, from q. This explains why p must be R-nested. On the other hand, if a certain number of iterations, say D, have passed since the start of this propagation and p has still remained stabilized during all that time, it is highly likely that the new messages did not actually affect that node.

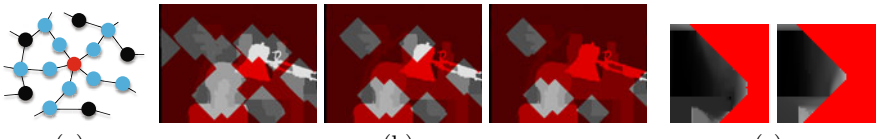


Fig. 2. (a) The red node is 2-nested, if itself and all blue nodes are stabilized. **(b)** Distribution of fixed nodes (red pixels) at 3 different iterations. **(c)** The same part of the 'confidence' map at 2 iterations of Tsukuba. More fixed nodes exist in the right map, which results into some non-fixed nodes becoming more 'confident' (*i.e.*, brighter).

As the dual-based algorithm progresses towards convergence, more and more nodes become fixed. This results into significant computational savings per iteration as only a very small number of dual variables have to be updated, which in turn results into a larger rate of improvement of the dual objective per iteration and thus in faster convergence. Fig. 2(b) shows examples from the distribution of the fixed nodes at different iterations of the multigrid algorithm for Tsukuba. Notice the order by which nodes become fixed: 'easier' nodes fix their labels earlier, while 'uncertain' nodes are fixed towards the end.

Another very important advantage of the decimation strategy is that, by fixing some of the labels, it manages to propagate additional information into the graph, which further increases the rate of improvement of the dual. This was found to considerably speed up convergence in our experiments. This propagation is illustrated by the 'confidence' maps for the 'tsukuba' example in Fig. 2(c), which show that, as a result of the decimation process, the 'confidence' of non-fixed nodes increases as well. Note that the confidence of a node p is calculated

by computing for each label the sum of its min-marginals for all the slave MRFs containing p and taking the difference between the two lowest sums.

But how can we empirically test the soundness of the above decimation process? A very strong empirical indication comes from the following fact: let us assume that the original dual LP relaxation is tight (or almost tight), *i.e.*, the resulting labels are (almost) optimal, which is the main case of interest. Note that each time we fix the label of a node, we are essentially modifying that relaxation. Moreover, the optimum of the modified dual relaxation increases only whenever a newly fixed node is assigned a suboptimal label. Therefore, in this case we can check how well the decimation process performed by simply comparing the original dual optimum with the dual optimum of the modified relaxation that results from fixing all the nodes. In all the real examples that we have tried, the two dual optima were either exactly the same (when the original relaxation was exact) or differed by a very small amount (when the original relaxation was almost tight). We have also verified this property with experiments on synthetic problems. Moreover, the obtained MRF energies were always better than the ones of the full algorithm (we found no case where this was not true).

Intuitively, the reason that we are able to obtain better primal solutions is because, by fixing some of the labels, we implicitly manage to gradually tighten the relaxation. Typically, LP-based solvers for MAP estimation function by solving the LP and then rounding each node to generate an integer solution. Instead, a better approach would be that, after rounding each node, we add its fixed state as an additional constraint to the LP and then solve this new LP before rounding the next node. This second approach, however, is very expensive but gives better solutions as the LP guiding the rounding scheme gets progressively tighter. The proposed decimation strategy can be thought of as an efficient way to approximately perform such an expensive series of computations. Stable nodes will have the same reparameterization in the final stage of the LP as they do now. Therefore, they can be immediately rounded, and their new solution propagated as a constraint. Note that the benefit of a decimation process to solving difficult problems has also been observed in other cases as well, *e.g.*, for solving SAT instances using survey propagation [8].

8 Extensions

Higher order MRFs: Due to the generality of the proposed formulation, the "algebraic multigrid" approach can also be extended to higher-order MRF optimization problems. These problems have the following form:

$$\mathrm{MRF}_G(\mathbf{U}, \mathbf{H}) := \min_{\mathbf{x}} \sum_{p \in \mathcal{V}} U_p(x_p) + \sum_{c \in \mathcal{C}} H_c(\mathbf{x}_c) \ , \tag{17}$$

where $\mathbf{H} = \{H_c\}$ are the higher order potential functions, which are now defined on cliques $c \in \mathcal{C}$ and replace the pairwise potentials $\mathbf{P}$.

Therefore, the dual objective function (2) now involves higher order potentials $\mathbf{H}$ (instead of $\mathbf{P}$), while the slave MRFs are defined on sub-hypergraphs G_i of

a hypergraph G [14]. The projection $\text{proj}(G)$ of any hypergraph G is defined analogously to the projection of a graph, *i.e.*, as the projection of its cliques. Similarly, the projection of an MRF with higher potentials $\mathbf{H}$ gives rise to an MRF with higher potentials $\bar{\mathbf{H}}$, which are again defined analogously to (6), *i.e.*,

$$\bar{H}_{\bar{c}}(\cdot) = \sum\nolimits_{c:\text{proj}(c)=\bar{c}} H_c(\cdot) \ . \tag{18}$$

Hence, by replacing $\mathbf{P}$ and $\bar{\mathbf{P}}$ with $\mathbf{H}$ and $\bar{\mathbf{H}}$ respectively, the restriction and prolongation operators PROJ and LIFT can then be computed using exactly the same algorithms as described in sections 5 and 6.

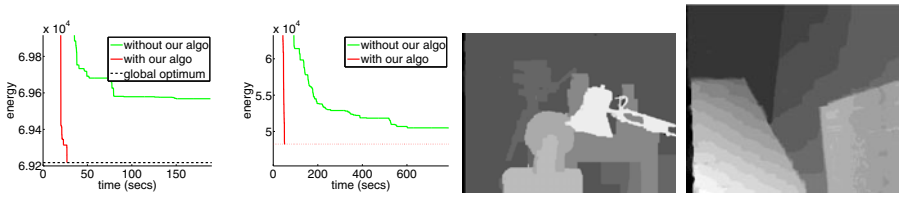
Tighter LP relaxations: In the dual decomposition framework, a tighter dual relaxation can result simply by choosing a set of non tree-structured slave MRFs. For instance, one can use loopy subgraphs of small tree-width for this purpose (intuitively, such a relaxation is tighter because the slaves now have higher optimal energies, and thus lead to better lower bounds). As a result, exactly the same algrebraic multigrid framework can be applied, thus leading to a multiresolution set of tighter relaxations in this case.

Data-driven projections: Typically the partitions that determine each projection in the hierarchy are chosen a priori (*e.g.*, for grids, a node at one level can project to a block of nodes at a coarser level). However, due to the generality of the proposed formulation, this could very well not be the case. Instead, one can use data driven partitions for defining these projections. In vision problems, for instance, it would be very useful to define these partitions so as to align with some of the edges in the image. If chosen properly, such data driven projections can lead to even greater computational savings.

9 Experimental Results

We have applied our method to a wide variety of vision problems. We first report results on pairwise MRFs. To this end, we tested our algorithm on the Middlebury dataset [5], which contains a variety of MRF problems on stereo matching, image segmentation and image denoising (all MRF potentials were set exactly the same as in that dataset). To demonstrate our framework for pairwise MRFs, we have used it to improve the TRW-S algorithm [2], which is a popular dual LP-based method for pairwise energies. We thus report results when we apply that algorithm with and without our framework. In both cases we use the same implementation of TRW-S as well as the same set of settings.[5] Slaves were chosen to be trees, with one tree per horizontal and vertical line of the input grid structured graph. We show typical plots of how the energy varies in Figs. 3(a),3(b) and the corresponding solutions in Figs. 3(c),3(d). Notice the much faster convergence when our framework is used. Further running times and energies for problems from the middlebury dataset are reported in Fig. 4. As

[5] For completeness we also compared our method with the original implementation of TRW-S by V. Kolmogorov (see the supplemental material [12] for these results).



(a) Tsukuba energies (b) Venus energies (c) (d)

Fig. 3. Convergence plots and results for Tsukuba and Venus

can be seen, our method provides a very significant speedup in all cases, while at the same time it increases the effectiveness of the optimization. In fact, it always computed solutions whose energy was lower than the best energy reported in the Middlebury dataset. This behaviour was consistent throughout all our experiments. For instance, for the 'tsukuba' example, our method computed the global optimum in a time that was at least an order of magnitude faster than the method in [15] (global optimality can be verified based on the dual lower bounds). For obtaining these results, we used an MRF hierarchy consisting of 3-5 levels, where the partition at each level was consisting of sets of 2×2 pixels. Also, parameters R and D (used in the decimation strategy) were set to some reasonably large values (*e.g.*, $R \geq 30$ and $D \geq 10$ on average).

We also tested our method on problems with higher order MRFs. To this end, we applied it to image segmentation and stereo matching problems, where we used a $\mathcal{P}^n$ Potts model [16] and a truncated second order derivative as higher order potentials, respectively. Both of them were solved using the framework of pattern-based potentials from [14]. We report indicative energies and running times for two such cases in Fig. 5(a), while Fig. 5(b) shows the corresponding result for stereo matching. As can be observed, even for high order MRF problems, our framework enables us to obtain high quality solutions much faster. It also increases the effectiveness of the optimization, as it still consistently leads to solutions of lower energy even in this case.

Finally, for completeness, we also compare our algorithm to the algorithm from [6] that uses BP in conjunction with a geometric multigrid method. Fig. 5(c)

	Without our algo		With our algo		
problem	energy	time(secs)	energy	time(secs)	speedup
Tsukuba	69568	188.53	69218*	26.93	7.00x
Venus	50392	795.11	48255	51.87	15.58x
Teddy	44505	1052.10	38299	131.25	8.00x
Flower	11274	2.74	11274*	0.46	5.95x
Sponge	27165	3.33	27165*	0.49	6.79x
Person	04852	5.94	04852*	0.65	9.13x
Penguin	54664	1016.30	48787	133.08	7.63x

* = global optimum

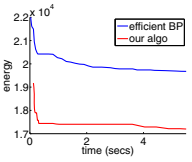
Fig. 4. Energies and running times for MRFs from the Middlebury dataset with and without our framework (energies have been normalized by subtracting a constant)

problem	Without our algo		With our algo		
	energy	time(secs)	energy	time(secs)	speedup
Segment (P^n potts)	325603	3.49	325603*	0.79	4.40x
Stereo (cones)	120840	181.27	118393	61.87	2.93x

* = global optimum

(a)

(b)



(c)

Fig. 5. (a) Energies and running times for high order MRFs. **(b)** Disparity for 'cones'. **(c)** Comparison between our method and the method in [6].

shows the convergence of the energy when these two algorithms are run on the stereo example from [6]. As can be seen, although the BP algorithm is very fast, our method computes a solution of lower energy even faster.

10 Conclusions

A framework for significantly improving the overall efficiency and effectiveness of dual-LP based methods was proposed in this paper, which is currently one of the main challenges encountered in energy minimization problems for vision. It relies on an algebraic multigrid approach and an efficient decimation strategy. It is also extremely general, and can be applied to both pairwise and higher order MRF problems. Due to this fact, and the very wide applicability of dual-LP based methods, we hope that our framework will help in making such methods much more practical for a wider class of vision problems in the future.

References

1. Wainwright, M., Jaakkola, T., Willsky, A.: Map estimation via agreement on trees: message-passing and linear programming. IEEE Trans. on Info. Theory (2005)
2. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. PAMI (2006)
3. Komodakis, N., Paragios, N., Tziritas, G.: MRF optimization via dual decomposition: Message-passing revisited. In: ICCV (2007)
4. Werner, T.: A linear programming approach to max-sum problem: A review. PAMI (2007)
5. Szeliski, R., et al.: A comparative study of energy minimization methods for markov random fields with smoothness-based priors. PAMI (2008)
6. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. IJCV 70, 41–54 (2006)
7. Komodakis, N., Tziritas, G.: Image completion using efficient belief propagation via priority scheduling and dynamic pruning. In: IEEE TIP (2007)
8. Braunstein, A., Mézard, M., Zecchina, R.: Survey propagation: An algorithm for satisfiability. Random Struct. Algorithms 27, 201–226 (2005)
9. Kovtun, I.: Partial optimal labeling search for a np-hard subclass of (max,+) problems. In: Michaelis, B., Krell, G. (eds.) DAGM 2003. LNCS, vol. 2781, pp. 402–409. Springer, Heidelberg (2003)

10. Alahari, K., Kohli, P., Torr, P.: Reduce, reuse and recycle: Efficiently solving multi-label mrfs. In: CVPR (2008)
11. Shekhovtsov, A., Kovtun, I., Hlaváč, V.: Efficient mrf deformation model for non-rigid image matching. In: CVIU (2008)
12. http://www.csd.uoc.gr/~komod/publications/docs/eccv10_supp.pdf
13. Dykstra, R.L.: An iterative procedure for obtaining i-projections onto the intersection of convex sets. Annals of Probability (1985)
14. Komodakis, N., Paragios, N.: Beyond pairwise energies: Efficient optimization for higher-order MRFs. In: CVPR (2009)
15. Meltzer, T., Yanover, C., Weiss, Y.: Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation. In: ICCV (2005)
16. Kohli, P., Kumar, P., Torr, P.: P3 and beyond: Solving energies with higher order cliques. In: CVPR (2007)

Energy Minimization under Constraints on Label Counts

Yongsub Lim[1], Kyomin Jung[1,⋆], and Pushmeet Kohli[2]

[1] Korea Advanced Institute of Science and Technology, Daejeon, Korea
yongsub@kaist.ac.kr, kyomin@kaist.edu
[2] Microsoft Research, Cambridge, United Kingdom
pkohli@microsoft.com

Abstract. Many computer vision problems such as object segmentation or reconstruction can be formulated in terms of labeling a set of pixels or voxels. In certain scenarios, we may know the number of pixels or voxels which can be assigned to a particular label. For instance, in the reconstruction problem, we may know size of the object to be reconstructed. Such label count constraints are extremely powerful and have recently been shown to result in good solutions for many vision problems.

Traditional energy minimization algorithms used in vision cannot handle label count constraints. This paper proposes a novel algorithm for minimizing energy functions under constraints on the number of variables which can be assigned to a particular label. Our algorithm is deterministic in nature and outputs ε-approximate solutions for all possible counts of labels. We also develop a variant of the above algorithm which is much faster, produces solutions under almost all label count constraints, and can be applied to all submodular quadratic pseudo-boolean functions. We evaluate the algorithm on the two-label (foreground/background) image segmentation problem and compare its performance with the state-of-the-art parametric maximum flow and max-sum diffusion based algorithms. Experimental results show that our method is practical and is able to generate impressive segmentation results in reasonable time.

1 Introduction

Algorithms for energy minimization have become an indispensable tool in computer vision. These algorithms enable inference of the Maximum a Posteriori (MAP) solutions of labelling problems such as image segmentation, optical flow, and stereo. Due to its wide applicability, the energy minimization problem has received a lot of interest from both the theoretical computer science [1] and machine learning communities [2,3,4,5].

Most energy minimization methods used in computer vision such as Max-product Belief Propagation (BP) [6,7], Tree Reweighted message passing (TRW) [8], and Graph Cuts [9] operate on energy functions defined over discrete variables. They consider unconstrained minimization of the energy function over the discrete domain of

⋆ This work was supported by the Engineering Research Center of Excellence Program of Korea Ministry of Education, Science and Technology(MEST) / National Research Foundation of Korea(NRF) (Grant 2009-0063242).

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 535–551, 2010.

random variables, and do not allow any direct way of enforcing constraints on the solutions. In contrast, this paper studies the problem of constrained energy minimization. Specifically, we address the problem of minimizing energy functions under the constraint that the number of variables assigned on any given label is equal to some known constant. This constrained minimization problem is useful for many labelling problems in computer vision. For instance, in the image segmentation problem, it enables us to obtain a segmentation of any desired size.

Related Work. During the past few years, researchers have considered practical optimization problems under relevant constraints. One of the most effective examples is the case of 3D reconstruction, where the silhouette constraint was introduced [10,11]. This constraint ensured that a ray emanating from any silhouette pixel must pass through one voxel which belongs to the 'object'. It was proven to be an effective replacement for the ballooning term [12] and led to improved results.

Segment *connectivity* is another example of an equally powerful but much more sophisticated constraint that was introduced for the two label (foreground/background) segmentation problem. The energy function corresponding to the segmentation problem is composed of unary and pairwise potential functions [13] and is well known to be *submodular*. This property allows the energy function to be minimized in polynomial time using efficient maximum flow based algorithms. The connectivity constraint enforces that all variables that have been assigned in the foreground label form one connected component. Vicente *et al.* [14] showed that enforcing connectivity while minimizing the submodular segmentation energy makes the problem NP-hard.

Constraints on Label Counts. Minimization under the so-called *label counting* constraints is not new to the computer science community. Unconstrained energy minimization was studied in theoretical computer science in the context of Metric labelling. This is the problem of minimizing an energy function where the pairwise potential functions are defined in terms of the weighted uniform distance function that is defined on the label set $[k]$. Recently, Naor and Schwartz [15] obtained an approximation algorithm for a constrained version of this problem, which they called the *balanced metric labeling problem*.

Balanced labeling means that the number of variables assigned to any particular label is at most ℓ. They obtain an $O\left(\frac{\log n}{\varepsilon}\right)$-approximation randomized algorithm that runs in polynomial time over n and $\frac{1}{\varepsilon}$. The algorithm guarantees that at most $O\left(\log k \frac{1+\varepsilon}{1-\varepsilon}\right) \cdot \ell$ many variables in the final solution are assigned to each label. The Naor-Schwartz algorithm works for any underlying graph G, but it cannot be applied for general fixed label counting constraints. Due to randomness of the assignment, the counting guarantee from their method is still far from the exact counting constraint we want to achieve. Furthermore, the approximation ratio between its answer and the optimal solution is not small enough to be useful in practice.

Counting Constraints in Computer Vision. Werner [16] were one of the first to introduce constraints on label counts in energy minimization. They proposed a n-ary max-sum diffusion algoithm for solving these problems, and demonstrated its performance

on the binary image denoising problem. However, their algorithm could only produce solutions or some label counts. It was not able to guarantee an output for any arbitrary label count desired by the user.

A number of other recent vision papers have also demonstrated how knowledge about label counts can be used as a useful prior. For instance, Woodford *et al.* [17] recently showed how potentials for enforcing a particular distribution in label counts can be used to improve results in labelling problems such as image denoising and texture synthesis. They proposed a number of sophisticated algorithms which were able to minimize energy functions containing higher order potentials encouraging particular counts of labels. However, their algorithms were not able to enforce label counts as a hard constraint, and also lacked any worst-case bounds on the quality of the obtained solution.

The method most closely related to ours is that of Kolmogorov *et al.* [18]. They showed that for submodular energy functions, the parametric maxflow algorithm [19] can be used for energy minimization with label counting constraints. However, this algorithm outputs optimal solutions for only some label counts, and is not guaranteed to output solutions for any arbitrary count of labels.

Our Results. We propose a new method for performing energy minimization under constraints on the label counts. Our algorithm is deterministic in nature and outputs ε-approximate solutions in a grid graph with N vertices. For all possible labels, the algorithm runs in $O\left(Nk^{\frac{1}{\varepsilon}}\left(\frac{1}{\varepsilon}\right)^{2k+2} + N^k\left(\frac{1}{\varepsilon}\right)^2\right)$ time, where k is the number of labels and N is the number of pixels. This algorithm can also minimize energy functions containing potentials depending on label counts such as the ones use in [17] as it outputs the minimum energy solution under all possible label counting constraints.

We also develop a variant of the above algorithm which is much faster and produces solutions under almost all label count constraints, but can only be applied to submodular quadratic pseudoboolean functions. We call this algorithm *decomposed parametric*. It is inspired from the parametric maxflow based method for obtaining solutions under label counts. As mentioned earlier, the vanilla parametric maxflow method finds optimal solutions for a small number of label counts. We propose a new algorithm which dramatically increases the number of label counts for which a solution can be found. We first decompose the original image into a number of subimages. The vanilla parametric maxflow algorithm is run on each subimage. In this way, we obtain a set of assignments for each sub-image with minimum energy under some label counts. These sets are merged to obtain the set of assignments for the whole image with minimum energy under all label count constraints. Experiments on the binary image segmentation problem show that our method dramatically outperforms the standard parametric maxflow and max-sum diffusion based methods for obtaining solutions under label count constraints.

1.1 Organization

Remainder of the paper is organized as follows. We define the problem setup and provide some preliminaries in section 2. In section 3, we state our main theorem about the multiplicative error bound guaranteed by our approach. Our parametric maxflow based

algorithm is explained in section 4. In section 5, we provide the results of our experiments on the image segmentation problem, and compare them with those obtained using state-of-the-art methods. We conclude by discussing ideas for future work in section 6.

2 Preliminaries and Setup

Energy Minimization. Many labeling problems in computer vision can be formulated using energy minimization. Energy functions are defined on a pixel-grid graph $G = (V, E)$, and have the form

$$H(\mathbf{x}) = \sum_{v \in V} \phi_v(x_v) + \sum_{(v,w) \in E} \phi_{vw}(x_v, x_w), \tag{1}$$

where $\phi_{vw} : [k]^2 \rightarrow \mathbb{R}^+ \stackrel{\triangle}{=} \{x \in \mathbb{R} : x \geq 0\}$ and $\phi_v : [k] \rightarrow \mathbb{R}^+$ are assumed to be arbitrary non-negative real-valued functions defined over variables taking values from the label set $[k] = \{1, \cdots, k\}$.

We use the positivity of ϕ_v's and ϕ_{vw}'s in the proof of our multiplicative approximation guarantee. However, our algorithm can also be applied to energy functions with negative ϕ_{vw} values in the same manner.

In this paper, we are interested in finding an assignment $\mathbf{x}$ minimizing H under a label count defined as follows.

Definition 1 (label count). *For an assignment* $\mathbf{x} \in [k]^N$ *and* $j \in [k]$, *define count*$(\mathbf{x}, j)$ *to be the number of variables* $x_v : v \in V$ *such that* $x_v = j$. *Let* $\mathcal{C}(N)$ *be the collection of* $C = (C_1, C_2, \ldots, C_k) \in {\mathbb{Z}_+}^k$ *such that* $C_1 + C_2 + \ldots + C_k = N$. *We call* $C \in \mathcal{C}(N)$ a label count. *For* $C \in \mathcal{C}(N)$, *let*

$$\mathcal{R}(C) = \{\mathbf{x} \in [k]^N | \forall j \in [k],\ \textit{count}(\mathbf{x}, j) = C_j\}.$$

Our problem is to find such an assignment $\mathbf{x}^*(C)$ that minimizes the energy function $H(\mathbf{x})$ among $\mathbf{x} \in \mathcal{R}(C)$. The problem of finding an assignment that minimizes energy for fixed label count even for a submodular $H(\mathbf{x})$ is known to be NP-hard [20]. Hence we consider the following approximation problem.

Definition 2 (ε-approximation). *Let* $0 < \varepsilon < 1$. *An assignment* $\hat{\mathbf{x}}$ *is called* ε-*approximation of the energy with the label count* C, *if* $\hat{\mathbf{x}} \in \mathcal{R}(C)$ *and*

$$(1 - \varepsilon) H(\hat{\mathbf{x}}) \leq H(\mathbf{x}^*(C)) \leq H(\hat{\mathbf{x}}).$$

Definition 3 (submodular function). *A pseudoboolean function* $g(x_1, x_2) : \{0, 1\}^2 \rightarrow \mathcal{R}$ *is submodular if the following holds.*

$$g(0, 0) + g(1, 1) \leq g(0, 1) + g(1, 0).$$

An energy function is called submodular if all its pairwise terms are submodular. If H is a submodular, an assignment with the minimum energy can be computed efficiently by the graph-cut algorithm [21]. Submodular energy functions are widely used for labeling problems in computer vision.

Parametric maxflow. Parametric maxflow algorithm [18] is known that it gives some $\mathbf{x}$'s minimizing H under some label counts and can be applied when x_v's are in $\{0, 1\}$ and H is submodular [22]. It deals with parameterized energy function rather than the original one.

Parametric maxflow

Let $G = (V, E)$ be an undirected graph. Parametric maxflow is to minimize energy function $H^\lambda(\mathbf{x})$ for parameter $\lambda \in I$ in the interval $I \in \mathbb{R}$ where

$$H^\lambda(\mathbf{x}) = H(\mathbf{x}) + \lambda \sum_{v \in V} x_v. \tag{2}$$

Lemma 1. *If an assignment* $\mathbf{x}$ *minimizes the energy function* H^λ *for some* λ, $H(\mathbf{x})$ *is minimum under the same label count as* $\mathbf{x}$.

Proof. Assume that $\mathbf{x}$ does not minimize H under the label count. Let $\mathbf{x}'$ minimize H under the same label count as $\mathbf{x}$. Because $\mathbf{x}$ and $\mathbf{x}'$ have same number of 1's, they also have the same second term in (2), therefore $H^\lambda(\mathbf{x}) > H^\lambda(\mathbf{x}')$. It contradicts that $\mathbf{x}$ minimizes H^λ.

For a fixed λ, note that H^λ is also submodular, hence (2) can be solved in polynomial time using the graph-cut algorithm. Because $F(\lambda) = \min_{\mathbf{x}}(H^\lambda(\mathbf{x}))$ is a piecewise-linear concave function of λ, it is enough to compute $\mathbf{x}$'s at all breakpoints of F rather than for every λ, where a breakpoint is the intersection point of two line segments of $F(\lambda)$. The following algorithm finds $\mathbf{x}$'s at each breakpoint and they are minimum of H under the label counts as that of $\mathbf{x}$.

Parametric maxflow algorithm

– Input : Energy function H.
1. Let $I = [\lambda_{min}, \lambda_{max}]$.
2. Compute $\mathbf{x}_{min}$ and $\mathbf{x}_{max}$ solutions for λ_{min} and λ_{max}, respectively.
3. **if** $\mathbf{x}_{min} = \mathbf{x}_{max}$ **then** Initialize L as $(\mathbf{x}_{min}, [\lambda_{min}, \lambda_{max}])$.
4. **else** Initialize L as $(\mathbf{x}_{min}, \{\lambda_{min}\})$, $(\mathbf{x}_{max}, \{\lambda_{max}\})$.
5. **while** there are adjacent items $(\mathbf{x}_i, I_i)$, $(\mathbf{x}_j, I_j)$ such that $\sup I_i < \inf I_j$
6. $\lambda_i = \sup I_i$, $\lambda_j = \inf I_j$.
7. Compute λ^*, a solution of $H^\lambda(\mathbf{x}_i) = H^\lambda(\mathbf{x}_j)$.
8. **if** $\lambda_i = \lambda^*$ **then** $I_j = I_j \cup [\lambda_i, \lambda_j]$.
9. **else if** $\lambda_j = \lambda^*$ **then** $I_i = I_i \cup [\lambda_i, \lambda_j]$.
10. **else** λ^* must be in (λ_i, λ_j).
11. Compute $\mathbf{x}^*$ minimizing $H^{\lambda^*}(\mathbf{x})$.
12. **if** $\mathbf{x}^* = \mathbf{x}_i$ **or** $\mathbf{x}^* = \mathbf{x}_j$
13. **then** $I_i = I_i \cup [\lambda_i, \lambda^*]$, $I_j = I_j \cup [\lambda^*, \lambda_j]$.
14. **else** $(\mathbf{x}, \{\lambda^*\})$ is inserted to L between $(\mathbf{x}_i, I_i)$ and $(\mathbf{x}_j, I_j)$

– Output : list L of pairs $(\mathbf{x}_i, I_i)$ where $H^\lambda(\mathbf{x}_i)$ is minimum for $\lambda \in I_i$.

This algorithm uses graph-cut algorithm at most $(2B + 2)$ many times where B is the number of breakpoints. In the worst case, there are at most $|V| + 1$ breakpoints since there is at most one break point for each label count. In the rest of this paper, we will call this parametric maxflow algorithm as *the pure parametric* algorithm (PP), to compare it with our new algorithms.

3 Approximation Algorithm for Energy Minimization

In this section, we provide an algorithm that is guaranteed to compute an ε-approximate solutions for all label counts in $O(|V|)$ time. Our algorithm can be generalized to more general class of graphs including 8-connected grid graph and planar graphs, by designing a collection of graph decompositions satisfying the properties of Lemma 2. For example, when the graph is a planar graph, the collection of decompositions in [23] satisfies the properties. In this paper, we will prove our result for grid graph for easy of explanation. Let G be the grid graph of size $n \times n$. Our algorithm is based on a decomposition of G into small components; computing an array of solutions in each of these components, then producing a global solution. The algorithm works by exploiting the sparseness of the graph G. It reduces the original problem with large tree-width to a number of smaller problems with low tree width. In this process, it inserts a error in the estimation. We have shown that this error is small because the graph has limited connectivity.

Theorem 1. *There is a deterministic algorithm that outputs ε-approximate solutions for all $C \in \mathcal{C}(N)$, which runs in time $O\left(Nk^{\frac{1}{\varepsilon}}\left(\frac{1}{\varepsilon}\right)^{2k+2} + N^k\left(\frac{1}{\varepsilon}\right)^2\right)$.*

We will call our algorithm DD (decomposed dynamic), since its main procedure is based on graph decompositions and dynamic programming on each graph components. A brief sketch of DD is as follows. First, we will obtain a family $\mathcal{D}$ of graph decompositions of G. It satisfies that each decomposition $D \in \mathcal{D}$ is obtained by removing some edges of G, and $|\mathcal{D}| = \frac{1}{\varepsilon^2}$. The following is a pseudo-code of our graph decomposition.

Graph Decomposition

- Inputs : $G = (V, E)$, $\varepsilon > 0$, and $k_1, k_2 \in \{0, 1, 2, \ldots, \frac{1}{\varepsilon}\}$.

1. Remove all the edges of G that connects vertices of the form (a, b) and $(a+1, b)$ where $a \equiv k_1 \pmod{\frac{1}{\varepsilon}}$.
2. Remove all the edges of G that connects vertices of the form (a, b) and $(a, b+1)$ where $b \equiv k_2 \pmod{\frac{1}{\varepsilon}}$.
3. The remaining graph is decomposed into connected components.

- Output : G.

Let $\mathcal{D}$ be the collection of the decompositions computed for all $k_1, k_2 \in \{0, 1, 2, \ldots, \frac{1}{\varepsilon}\}$.

Lemma 2. *$\mathcal{D}$ satisfies the following properties.*
(1) For all $D \in \mathcal{D}$, the size of each connected component of D is at most $\frac{1}{\varepsilon^2}$.
(2) For all edge e of G, the number of decompositions in $\mathcal{D}$ that remove e is at most $\varepsilon|\mathcal{D}|$.

Now, fix $D \in \mathcal{D}$. Let $R_1, R_2, \ldots, R_\ell$ be the connected components of D. For each $R_i = (V_i, E_i)$, DD computes the following g_i values for all $C_{(i)} = (C_{i1}, C_{i2}, \ldots, C_{ik}) \in \mathcal{C}(|V_i|)$ by dynamic programming on R_i.

$$g_i(C_{(i)}) = \min_{\mathbf{x} \in \mathcal{R}(C_{(i)})} \left[\sum_{v \in V_i} \phi_v(x_v) + \sum_{(v,w) \in E_i} \phi_{vw}(x_v, x_w) \right].$$

Note that the tree width of the subgraph R_i is at most $\frac{1}{\varepsilon}$. The description of computation of g_i for all $C_{(i)} \in \mathcal{C}(|V_i|)$ by dynamic programming is in Appendix II in the supplementary material.

Computation of each g_i for all $C_{(i)} \in \mathcal{C}(|V_i|)$ takes $O\left(k^{\frac{1}{\varepsilon}}\left(\frac{1}{\varepsilon}\right)^{2k}\right)$ time. Since there are at most $O(N)$ many R_i's, its total computation time for a fixed $D \in \mathcal{D}$ is $O\left(Nk^{\frac{1}{\varepsilon}}\left(\frac{1}{\varepsilon}\right)^{2k}\right)$.

Now, Let $E_D \subset E$ be the union of all the edges of R_i's. Then for each $C \in \mathcal{C}(N)$, we compute

$$g_D(C) = \min_{\mathbf{x} \in \mathcal{R}(C)} \left[\sum_{v \in V} \phi_v(x_v) + \sum_{(v,w) \in E_D} \phi_{vw}(x_v, x_w) \right],$$

using $g_i(\cdot)'s$. This can be done in time $O(N^k)$ by the merging process described below.

Merging

- We begin with the regions $R_1, R_2 \ldots R_\ell$, and their corresponding functions g_i's.
- Repeat the following process until there remains just one region.
  - If there are more than one regions, choose any two of them, say R_a and R_b. Let g_a and g_b be their corresponding functions.
  - Let $R_c = R_a \cup R_b$ in the sense of graph union. I.e, for $R_a = (V_a, E_a)$ and $R_b = (V_b, E_b)$, let $R_c = (V_c, E_c)$ with $V_c = V_a \cup V_b$ and $E_c = E_a \cup E_b$.
  - For each $C_{(c)} \in \mathcal{C}(|V_c|)$, let

$$g_c(C_{(c)}) = \min \left[g_a(C_{(a)}) + g_b(C_{(b)}) \right], \tag{3}$$

    where the minimization is over all $C_{(a)} \in \mathcal{C}(|V_a|)$, $C_{(b)} \in \mathcal{C}(|V_b|)$ such that $C_{ai} + C_{bi} = C_{ci}$ for all $1 \leq i \leq k$.
  - Replace the two regions R_a and R_b by R_c. Also replace g_a and g_b by g_c.
- Output the resulting function for the entire graph.

In the above process, under the assumption that for all $C_{(a)} \in \mathcal{C}(|V_a|)$,

$$g_a(C_{(a)}) = \min_{\mathbf{x} \in \mathcal{R}(C_{(a)})} \left[\sum_{v \in V_a} \phi_v(x_v) + \sum_{(v,w) \in E_a} \phi_{vw}(x_v, x_w) \right],$$

and that for all $(C_{(b)}) \in \mathcal{C}(|V_b|)$,

$$g_b(C_{(b)}) = \min_{\mathbf{x} \in \mathcal{R}(C_{(b)})} \left[\sum_{v \in V_b} \phi_v(x_v) + \sum_{(v,w) \in E_b} \phi_{vw}(x_v, x_w) \right],$$

it is straightforward to see that for all $C_{(c)}$,

$$g_c(C_{(c)}) = \min_{\mathbf{x} \in \mathcal{R}(C_{(c)})} \left[\sum_{v \in V_c} \phi_v(x_v) + \sum_{(v,w) \in E_c} \phi_{vw}(x_v, x_w) \right].$$

Hence by induction, we obtain that the above procedure outputs $g_D(\cdot)$, and its total running time is $O(N^k)$.

Let $\hat{\mathbf{x}}_D(C)$ be the assignment corresponding to $g_D(C)$. Then let

$$\hat{\mathbf{x}}(C) = \text{argmin}_{D \in \mathcal{D}} H(\hat{\mathbf{x}}_D(C))$$

be the output of DD for $C \in \mathcal{C}(N)$. Approximation proof of DD is in Appendix I.

4 Decomposed Parametric

Although DD is guaranteed to output ε-approximation for all label counts, it runs slowly when the decomposed subimage size becomes large. In this section we provide a more practical algorithm that works for any submodular pseudoboolean energy function.

Our new algorithm, DP (decomposed parametric) runs on a decomposed image like in DD. The basic idea is to apply the parametric maxflow to each decomposed subimage rather than the dynamic programming. While DD outputs optimal assignments under all label counts in each subimage, DP outputs some of those assignments. However, by merging the partially optimal results of parametric maxflow, we obtain assignments under almost all label counts. Although our algorithm can be applied to an image of arbitrary size, to make explanation easy, we assume a square size image in this section. Note that since DP uses the parametric maxflow on each subimage, it is only applicable to binary labeling with submodular while DD can be applied to general labeling with any energy function.

DP decomposes a given $n \times n$ image to $\lceil \frac{n}{m} \rceil^2$ many subimages of size $m \times m$. Let I_{ij} be the subimage $[(i-1) \times m + 1, i \times m] \times [(j-1) \times m + 1, j \times m]$ and H_{ij} be the energy function H restricted to I_{ij} where $1 \leq i, j \leq \lceil \frac{n}{m} \rceil$. Figure 1 depicts this decomposition. We apply parametric maxflow algorithm to every subimage I_{ij} to compute assignments minimizing H_{ij} under some label counts. Here we assume that the output of the pure parametric is a form of an array A such that $A[k]$ is an assignment having label count k. Then arrays of size $m^2 + 1$ are created as results of the parametric maxflow algorithm on each subimage, and we obtain an array about the whole image by merging those arrays.

Decomposed Parametric

– Inputs : an image I of size $n \times n$, an integer m.
1. $A = \emptyset$.
2. Decompose I to subimages of size $m \times m$.
3. **for** i 1 to $\lceil n/m \rceil$
4. **for** j 1 to $\lceil n/m \rceil$
5. A_{ij} = *Parametric maxflow*(H_{ij}).
6. A = *Merge*(A, A_{ij}).

– Output : A.

Merge

– Inputs : two arrays A_1 and A_2
1. Let n_1 and n_2 be the size of A_1 and A_2, respectively.
2. Let A be a new array of size $n_1 + n_2 - 1$.
3. Every element of A is set to *empty* assignment.
4. **for** every $(i, j) \in \{0, \cdots, n_1 - 1\} \times \{0, \cdots, n_2 - 1\}$
5. **if** both $A_1[i]$ and $A_2[j]$ are not *empty* assignment
5. $\mathbf{x} = A_1[i]$ concatenated by $A_2[j]$.
6. **if** $H(\mathbf{x}) < H(A[i + j])$
7. $A[i + j] = \mathbf{x}$

– Output : A.

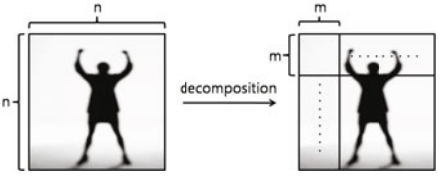


Fig. 1. Decomposition of a $n \times n$ image to subimages of size $m \times m$

When two arrays are merged, we get $\ell_1 \times \ell_2$ new assignments where ℓ_1 is the number of assignments in the first array and ℓ_2 is that in the second array. Although there are some overlap of label counts, the new array has bigger proportion of *nonempty* assignments than the two merged arrays. We observe that when m is about $\frac{n}{3}$, for almost all label counts DP outputs assignments.

5 Experimental Results

We did experiments to compare our two algorithms with PP. To that end, we measured three values: the number of label counts for which the methods returned a solution, the energy values for the computed assignments, and the running time.

For our image segmentation experiments, we considered the energy function H defined in (1). The potential functions ϕ_i of H are obtained using the method described in [24] which exploits user given hints about the appearance of foreground and background segments. The pairwise potentials defined over edges connected in a 4-neighbourhood systems are defined as

$$\phi_{ij} = |x_i - x_j|(\lambda_1 + \lambda_2 \times g(i, j)),$$

where λ_1 and λ_2 are parameters of the model, and $g(i, j)$ is proportional to the distance of i and j's RGB colors. We have conducted experiments for various values of λ_1 and λ_2. In our experiments, we use DP_k and DD_k to denote DP and DD , respectively, with the decomposition of an image into $k \times k$ number of subimages. The parameter values used in the experiments are specified by $\mu = \frac{\lambda_2}{\lambda_1}$.

Table 1. Comparison of the number of label counts for the experiment 1. Each value is the average over 8 images. The ratio of the number of label counts for which each algorithm computes a solution over the number of possible label counts. For $\mu = 10$ and $\mu = 30$, the results are almost the same as those with $\mu = 20$. DP_3 outputs solutions for almost all label counts regardless of μ.

	$\mu = 20$		
λ_1	PP	DP_3	DP_5
1	0.2786	0.9828	0.9998
2	0.2552	0.9819	0.9997
4	0.2231	0.9795	0.9995
8	0.1862	0.9767	0.9994
16	0.1498	0.9736	0.9986
32	0.1164	0.9698	0.9970
64	0.0875	0.9650	0.9951
128	0.0642	0.9544	0.9925

Experiment 1. Our first experiment compares the average number of assignments minimizing the energy function H under some label counts, and the average energy values of DP with optimal solutions obtained by PP. We used 8 images from [25] for computing the average each of which was a 300×300 size, and simulated DP_k, $1 \leq k \leq 5$, and PP for all images. For each algorithm, we did tests for $\lambda_1 = 2^i$, $0 \leq i \leq 7$, and $\mu = 10, 20$ and 30.

Experimental results are shown in Table 1 and Figure 2. Table 1 shows the average ratio of the number of computed label counts over N, the number of pixels of the image. DP_3 outputs assignments under almost all label counts while PP about 20% of the label counts. DP_5 outputs assignments under label counts more than 99% regardless of λ_1 and λ_2.

Figure 2 shows the average energy value ratio of each simulation compared to that of the optimal solutions obtained by PP . In Figure 2, the energy value of DP over the optimal energy tends to have bigger error as λ_1 or λ_2 increases, or the image is decomposed to more subimages. But in almost all cases, the error is quite small, especially when $\lambda_1 = 8$, which is typically used in the image segmentation, the error is less than 0.5% regardless of λ_2. In short, we obtain the results that DP_3 is enough to obtain assignments under almost all label counts with only small error.

Table 2 shows the running time of PP and DPs. Note that DP is quite fast and its running time is competative to that of PP .

Examples of segmented images of the algorithms, PP and DP, are shown in Figure 3. Note that the output image of DP_3 is almost the same to that of PP. By comparing with the ground truth, we observe that DP_3 is nearly optimal.

We note that adding a constant to the energy function affects the approximation ratio of the results. In our simulation, we adjusted the constant term for each vertex v so that one of $\phi_v(0)$ and $\phi_v(1)$ becomes 0.

Experiment 2. The second setup is for comparing the average energy values of DD, DP and PP. This experiment used 8 images each of which was a 200×200 size image, and simulated DP_{20} and DD_{20}. λ_1 and μ were the same as in the setup 1.

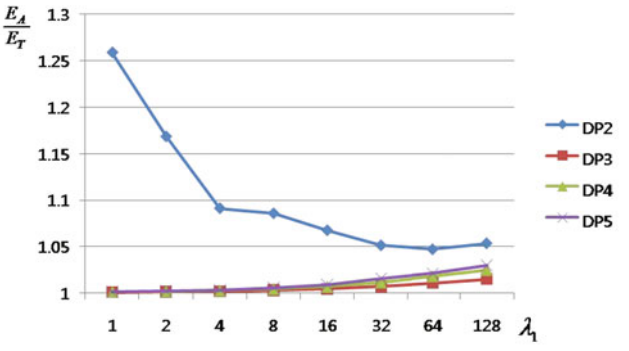
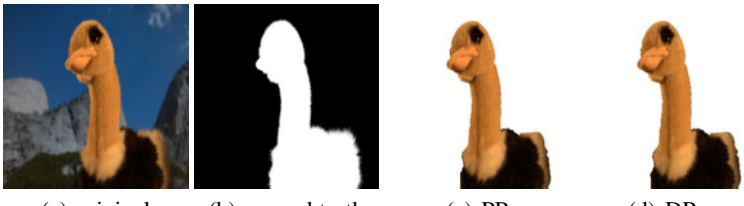


Fig. 2. Comparison of the average energy value for the experiment 1. We computed the average of $\frac{E_A}{E_T}$ over the computed label counts where E_A is the average energy value for 8 images of DP , and E_T is the average energy value of PP for 8 images. μ did not affect the values, so in this graph the values with $\mu = 20$ are shown.

Table 2. Table for running time of the algorithms for 8 images

	Time(seconds)				
	PP	DP_2	DP_3	DP_4	DP_5
IM1	8	10	18	24	27
IM2	11	23	31	42	53
IM3	51	22	28	35	42
IM4	4	11	15	18	19
IM5	23	42	48	63	75
IM6	12	38	42	52	62
IM7	26	44	55	76	94
IM8	17	42	46	58	70



(a) original (b) ground truth (c) PP (d) DP_3

Fig. 3. (b) is the ground truth for segmentation, (c) is the output of PP for a label count close to that of the ground truth, and (d) is the output of DPs with similar label count

Table 3 shows the average ratio of the energy values of DP_{20} and DD_{20} over the optimal solutions obtained by PP. It is similar with Figure 2 to increase the energy value as λ_1 or μ increases. Note that DP_{20} outputs almost the same energy ratio values with DD which is optimal in each decomposed subimages. Sometimes DP is even slightly

Table 3. Comparison of the average energy value for the experiment 2. We computed the values in the same way with Figure 2

	$\mu = 10$		$\mu = 20$		$\mu = 30$	
λ_1	DP_{20}	DD_{20}	DP_{20}	DD_{20}	DP_{20}	DD_{20}
1	1.0061	1.0079	1.0080	1.0104	1.0100	1.0127
2	1.0086	1.0109	1.0111	1.0140	1.0136	1.0172
4	1.0129	1.0157	1.0163	1.0199	1.0194	1.0235
8	1.0197	1.0232	1.0249	1.0295	1.0302	1.0359
16	1.0308	1.0360	1.0388	1.0456	1.0472	1.0558
32	1.0481	1.0492	1.0592	1.0611	1.0711	1.0739
64	1.0706	1.0713	1.0862	1.0877	1.1020	1.1044
128	1.1008	1.1021	1.1228	1.1253	1.1447	1.1484

better. This is because although DD is actually optimal in each subimage, after merging, it is not guaranteed to have lower energy than the output of the DP on the whole image.

Table 4 shows the running time of DD_{20} and DD_{25} for each image. Although DD is guaranteed to output approximate solutions, its running time is comparably longer than that of DP.

Table 4. Table for running time of DD_{20} and DD_{25} for 8 images

	Time	
	DD_{20}	DD_{25}
IM1	24m 41s	17m
IM2	32m 33s	29m 21s
IM3	24m 16s	16m 21s
IM4	25m 17s	17m 43s
IM5	27m 39s	21m 14s
IM6	26m 26s	19m 14s
IM7	33m 26s	25m 15s
IM8	27m 12s	22m 6s

From experiment 1 and 2, we can conclude that for binary labeling with submodular energy function, DP_3 performs best among PP , DP_i, DD when considering the number of label counts, accuracy, and the running time altogether.

Experiment 3. In this experiment, we compare our algorithm with the Werner's max-sum diffusion algorithm [16] on the binary image denoising problem. A binary image corrupted with Gaussian noise of size 150×150 is used. DP_3 with D_3, $\lambda_1 = 8$ and $\lambda_2 = 160$ and Werner's were simulated. Figure 4 shows the original image and two resulting images with the same label count. It can be seen that the DP method produced assignments for many more label count constraints (21568) while still remaining close to the ground truth result. In contrast, Werner's max-sum diffusion algorithm was only able to find solutions for 12 label counts.

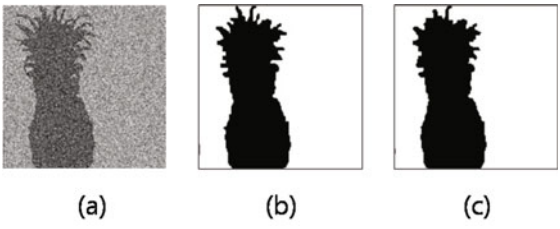


Fig. 4. (a) Original image of size 150×150 corrupted with Gaussian noise. (b) Result of DP with D_3 under label count 7058 among 21568 many outputs. (c) Result of Werner's max-sum diffusion algorithm under the same label count among 12 many outputs.

6 Discussion and Future Work

We have proposed novel algorithms for minimizing energy functions under label counting constraints. We first provided an efficient algorithm that outputs ε-approximate solutions for all possible counts of labels for any energy function. We also developed a variant of this algorithm which can be applied to submodular energy functions, that is much faster but misses solutions corresponding to some label counts.

In this paper, we have only considered the counting constraint defined on *vertices*. Another important counting constraint problem is that of *edge counting*, ie. the number of times we see a discontinuity in the labelling which is exactly equal to the boundary length in the case of image segmentation.

Consider the energy minimization with constraint on the number of *boundary edges* (edges having different labels on its two end vertices). This problem corresponds to segmentation with fixed boundary length. Note that this problem can be partially solved by considering the following parametric maxflow:

$$H^{\lambda}(\mathbf{x}) = H(\mathbf{x}) + \lambda \sum_{(v,w)\in E} x_v x_w,$$

where $\lambda \leq 0$ using the same reasoning as vertex counting. However, the algorithm is partial, since it cannot work for $\lambda \geq 0$ where the energy become non-submodular. As a future work, we will work on analysis of this algorithm.

References

1. Chekuri, C., Khanna, S., Naor, J., Zosin, L.: A linear programming formulation and approximation algorithms for the metric labelling problem. SIAM Journal on Discrete Mathematics (2005)
2. Komodakis, N., Tziritas, G., Paragios, N.: Fast, approximately optimal solutions for single and dynamic MRFs. In: CVPR (2007)
3. Kumar, M.P., Koller, D.: MAP estimation of semi-metric MRFs via hierarchical graph cuts. In: UAI (2009)
4. Sontag, D., Meltzer, T., Globerson, A., Jaakkola, T., Weiss, Y.: Tightening LP relaxations for MAP using message passing. In: UAI (2008)

5. Werner, T.: A linear programming approach to max-sum problem: A review. PAMI (2007)
6. Weiss, Y., Yanover, C., Meltzer, T.: MAP estimation, linear programming and belief propagation with convex free energies. In: UAI (2007)
7. Yedidia, J., Freeman, W., Weiss, Y.: Generalized belief propagation. In: NIPS (2001)
8. Wainwright, M., Jaakkola, T., Willsky, A.: MAP estimation via agreement on trees: message-passing and linear programming. IEEE Transactions on Information Theory (2005)
9. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI (2001)
10. Kolev, K., Cremers, D.: Integration of multiview stereo and silhouettes via convex functionals on convex domains. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 752–765. Springer, Heidelberg (2008)
11. Sinha, S., Pollefeys, M.: Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. In: ICCV (2005)
12. Vogiatzis, G., Torr, P., Cipolla, R.: Multi-view stereo via volumetric graph-cuts. In: CVPR (2005)
13. Boykov, Y., Jolly, M.: Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In: ICCV (2001)
14. Vicente, S., Kolmogorov, V., Rother, C.: Graph cut based image segmentation with connectivity priors. In: CVPR (2008)
15. Naor, J., Schwartz, R.: Balanced metric labeling. In: STOC (2005)
16. Werner, T.: High-arity interactions, polyhedral relaxations, and cutting plane algorithm for soft contraint optimisation (map-mrf). In: CVPR (2008)
17. Woodford, O., Rother, C., Kolmogorov, V.: A global perspective on MAP inference for low-level vision. In: ICCV (2009)
18. Kolmogorov, V., Boykov, Y., Rother, C.: Application of parametric maxflow in computer vision. In: ICCV (2007)
19. Gallo, G., Grigoriadis, M., Tarjan, R.: A fast parametric maximum flow algorithm and applications. SIAM J. on Comput. 18, 30–55 (1989)
20. Garey, M., Johnson, D.S.: Computers and intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, New York (1979)
21. Goldberg, A., Tarjan, R.: A new approach to the maximum-flow problem. Journal of the Association for Computing Machinery (1988)
22. Kohli, P.: Minimizing dynamic and higher order energy functions using graph cuts (2007)
23. Jung, K., Shah, D.: Local algorithms for approximate inference in minor-excluded graphs. In: NIPS (2007)
24. Blake, A., Rother, C., Brown, M., Pérez, P., Torr, P.: Interactive image segmentation using an adaptive GMMRF model. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 428–441. Springer, Heidelberg (2004)
25. Rhemann, C., Rother, C., Rav-Acha, A., Sharp, T.: High resolution matting via interactive trimap segmentation. In: CVPR (2008)

Appendix I

Let $C \in \mathcal{C}(N)$ be fixed, and let

$$\mathbf{x}^* = \mathrm{argmin}_{\mathbf{x} \in \mathcal{R}(C)} H(\mathbf{x}),$$

be the optimal solution, and let $\hat{\mathbf{x}} = \hat{\mathbf{x}}(C)$ be the output of DD for C. For each $D \in \mathcal{D}$, denote $\hat{\mathbf{x}}_D = \hat{\mathbf{x}}_D(C)$. From the positivity of ϕ_{vw}, for all $D \in \mathcal{D}$,

$$\sum_{v \in V} \phi_v(x^*_v) + \sum_{(v,w) \in E_D} \phi_{vw}(x^*_v, x^*_w) \leq H(x^*). \tag{4}$$

From the minimality of $\hat{\mathbf{x}}_D$ in each connected components of D, for all $D \in \mathcal{D}$,

$$\begin{aligned} &\sum_{v \in V} \phi_v((\hat{\mathbf{x}}_D)_v) + \sum_{(v,w) \in E_D} \phi_{vw}((\hat{\mathbf{x}}_D)_v, (\hat{\mathbf{x}}_D)_w) \\ \leq &\sum_{v \in V} \phi_v(x^*_v) + \sum_{(v,w) \in E_D} \phi_{vw}(x^*_v, x^*_w). \end{aligned} \tag{5}$$

From (4), (5) and the definition of $\hat{\mathbf{x}}$,

$$\begin{aligned} &\sum_{D \in \mathcal{D}} \left[\sum_{v \in V} \phi_v(\hat{\mathbf{x}}_v) + \sum_{(v,w) \in E_D} \phi_{vw}(\hat{\mathbf{x}}_v, \hat{\mathbf{x}}_w) \right] \\ \leq &\sum_{D \in \mathcal{D}} \left[\sum_{v \in V} \phi_v((\hat{\mathbf{x}}_D)_v) + \sum_{(v,w) \in E_D} \phi_{vw}((\hat{\mathbf{x}}_D)_v, (\hat{\mathbf{x}}_D)_w) \right] \\ \leq &\sum_{D \in \mathcal{D}} \left[\sum_{v \in V} \phi_v(x^*_v) + \sum_{(v,w) \in E_D} \phi_{vw}(x^*_v, x^*_w) \right] \\ \leq &|\mathcal{D}| H(x^*). \end{aligned} \tag{6}$$

By the property (2) of Lemma 2, i.e. from the property that for each edge e of E, the number of decompositions in $\mathcal{D}$ that removes e is at most $\varepsilon|\mathcal{D}|$, we obtain that

$$\begin{aligned} &(1-\varepsilon)|\mathcal{D}| H(\hat{\mathbf{x}}) \\ \leq &\sum_{D \in \mathcal{D}} \left[\sum_{v \in V} \phi_v(\hat{\mathbf{x}}_v) + \sum_{(v,w) \in E_D} \phi_{vw}(\hat{\mathbf{x}}_v, \hat{\mathbf{x}}_w) \right] \end{aligned} \tag{7}$$

From (6) and (7), we have

$$(1-\varepsilon) H(\hat{\mathbf{x}}_D) \leq H(\mathbf{x}^*).$$

Appendix II

Computing g_i

- Let $V_i = \{(a,b)|a,b \in \{1,2,\ldots\frac{1}{\varepsilon}\}\}$ be the set of vertices of R_i.
- Order the elements of V_i by dictionary order, i.e., $(a_1,b_1) < (a_2,b_2)$ if $a_1 < a_2$ or, $a_1 = a_2$ and $b_1 < b_2$. Let $v_1, v_2, \ldots, v_{\frac{1}{\varepsilon^2}}$ be the vertices in that order.
- For $t = 0, 1, \ldots, \left(|V_i| - \frac{1}{\varepsilon}\right) = t^*$, let
  $B_t = \left\{v_{t+1}, \ldots, v_{t+\frac{1}{\varepsilon}}\right\}$.
  Let $V_{it} = \{v \in V_i \mid$ order of v is less than or equal to some vertex in $B_t\}$. Let E_{it} be the set of edges that connect two vertices of V_{it}.
- For each assignment $\hat{\mathbf{x}}^{B_t} \in [k]^{|B_t|}$ over B_t, and each $C_{(t)} = (C_{t1}, C_{t2}, \ldots, C_{tk}) \in \mathcal{C}(|V_t|)$, let

$$\mathcal{R}(\hat{\mathbf{x}}^{B_t}, C_{(t)}) = \mathcal{R}(C_{(t)}) \bigcap \left\{\mathbf{x} \in [k]^{|V_{it}|} \mid \mathbf{x}_v = \hat{\mathbf{x}}_v^{B_t}\ \forall\, v \in B_t\right\}.$$

  We will compute the following for $t = 0, 1, \ldots, t^*$).

$$\hat{g}_t(\hat{\mathbf{x}}^{B_t}, C_{(t)}) = \min_{\mathbf{x} \in \mathcal{R}(\hat{\mathbf{x}}^B, C_{(t)})} \left[\sum_{v \in V_{it}} \phi_v(\mathbf{x}_v) + \sum_{(v,w) \in E_{it}} \phi_{vw}(\mathbf{x}_v, \mathbf{x}_w)\right].$$

- For $t = 0$, note that $V_{i0} = B_0$. Hence we directly compute $\hat{g}_0(\hat{\mathbf{x}}^{B_0}, C_{(0)})$ for all $\hat{\mathbf{x}}^{B_0} \in [k]^{|B_0|}$ and $C_{(0)} \in \mathcal{C}(|V_{i0}|)$.
- For $t = 1, 2 \ldots t^*$,
  - For each t, let $B'_t = B_t \bigcup B_{t-1}$. For each $\hat{\mathbf{x}}^{B'_t} \in [k]^{|B'_t|}$, and $C_{(t)} \in \mathcal{C}(|V_t|)$ let

$$\mathcal{R}(\hat{\mathbf{x}}^{B'_t}, C_{(t)}) = \mathcal{R}(C_{(t)}) \bigcap \left\{\mathbf{x} \in [k]^{|V_{it}|} \mid \mathbf{x}_v = \hat{\mathbf{x}}_v^{B'_t}\ \forall\, v \in B'_t\right\},$$

    and compute

$$\hat{g}'_t(\hat{\mathbf{x}}^{B'_t}, C_{(t)}) = \min_{\mathbf{x} \in \mathcal{R}(\hat{\mathbf{x}}^{B'_t}, C_{(t)})} \left[\sum_{v \in V_{it}} \phi_v(\mathbf{x}_v) + \sum_{(v,w) \in E_{it}} \phi_{vw}(\mathbf{x}_v, \mathbf{x}_w)\right]$$

    by the relation

$$\begin{aligned}\hat{g}'_t(\hat{\mathbf{x}}^{B'_t}, C_{(t)}) &= \hat{g}_{t-1}\left(\left(\hat{\mathbf{x}}^{B'_t}\right)_{B_{t-1}}, C'_{(t)}\right) \\ &+ \phi_{v_{t+\frac{1}{\varepsilon}}}\left(\left(\hat{\mathbf{x}}^{B'_t}\right)_{v_{t+\frac{1}{\varepsilon}}}\right) \\ &+ \phi_{v_{t+\frac{1}{\varepsilon}}, v_{t+\frac{1}{\varepsilon}-1}}\left(\left(\hat{\mathbf{x}}^{B'_t}\right)_{v_{t+\frac{1}{\varepsilon}}}, \left(\hat{\mathbf{x}}^{B'_t}\right)_{v_{t+\frac{1}{\varepsilon}-1}}\right) \\ &+ \phi_{v_{t+\frac{1}{\varepsilon}}, v_t}\left(\left(\hat{\mathbf{x}}^{B'_t}\right)_{v_{t+\frac{1}{\varepsilon}}}, \left(\hat{\mathbf{x}}^{B'_t}\right)_{v_t}\right),\end{aligned}$$

where $C'_{tj} = C_{tj} - 1$ for $j \in [k]$ such that $\left(\hat{\mathbf{x}}^{B'_t}\right)_{v_{t+\frac{1}{\varepsilon}}} = j$, and $C'_{tj} = C_{tj}$ for other j's. In the above computation, when there is no edge between $v_{t+\frac{1}{\varepsilon}}$ and $v_{t+\frac{1}{\varepsilon}-1}$, the term $\phi_{v_{t+\frac{1}{\varepsilon}},v_{t+\frac{1}{\varepsilon}-1}}$ is not computed.

- For $\hat{\mathbf{x}}^{B_t} \in [k]^{|B_t|}$ and $C_{(t)} \in \mathcal{C}(|V_t|)$, compute

$$\hat{g}_t(\hat{\mathbf{x}}^{B_t}, C_{(t)}) = \min_{j\in[k]} \hat{g}'_t(\left(j, \hat{\mathbf{x}}^{B_t}\right), C_{(t)}).$$

– For each $C_{(i)} \in \mathcal{C}(|V_i|)$, output

$$g_i(C_{(i)}) = \min_{\hat{\mathbf{x}}^{B_{t^*}} \in [k]^{|B_{t^*}|}} \hat{g}_{t^*}(\hat{\mathbf{x}}^{B_{t^*}}, C_{(i)}).$$

A Fast Dual Method for HIK SVM Learning

Jianxin Wu*

School of Computer Engineering, Nanyang Technological University
jxwu@ntu.edu.sg

Abstract. Histograms are used in almost every aspect of computer vision, from visual descriptors to image representations. Histogram Intersection Kernel (HIK) and SVM classifiers are shown to be very effective in dealing with histograms. This paper presents three contributions concerning HIK SVM classification. First, instead of limited to integer histograms, we present a proof that HIK is a positive definite kernel for non-negative real-valued feature vectors. This proof reveals some interesting properties of the kernel. Second, we propose ICD, a deterministic and highly scalable dual space HIK SVM solver. ICD is faster than and has similar accuracies with general purpose SVM solvers and two recently proposed stochastic fast HIK SVM training methods. Third, we empirically show that ICD is not sensitive to the C parameter in SVM. ICD achieves high accuracies using its default parameters in many datasets. This is a very attractive property because many vision problems are too large to choose SVM parameters using cross-validation.

1 Introduction

Recently, the Histogram Intersection Kernel (HIK) has attracted a lot of attention in the computer vision community. The success of HIK can be attributed to at least two important factors:

- First, histograms are frequently used in solving vision problems. At the feature level, many visual descriptors are histograms of various image measurements, e.g., SIFT [11], HOG [5], CENTRIST [20], or histogram of LBP [15], just to name a few. At the image level, histogram is also a popular representation, e.g., color histogram [17] or bag of visual words.
- Second, it is shown that HIK, as a measure for comparing the similarity (or dissimilarity) of *two histograms*, achieves better performances in various machine learning tasks than other commonly used measures, e.g., l_2 distance or RBF kernel. HIK is shown to have higher accuracies in SVM classification [13,19], and in the clustering of *histograms* [19].

Recently HIK becomes even more attractive for its fast evaluation speed [13,19]. It is shown that

$$\sum_{i=1}^{n} c_i \kappa_{\mathrm{HI}}(\boldsymbol{q}, \boldsymbol{x}_i) \tag{1}$$

* The author is supported by the NTU startup grant.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 552–565, 2010.

can be computed in $O(d)$ steps, where $\{\boldsymbol{x}_i\}_{i=1}^n$ are a set of n d-dimensional histograms, $\boldsymbol{q}$ is a query histogram, and $\kappa_{\text{HI}}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{j=1}^d \min(x_j, y_j)$ is the histogram intersection kernel. Although the effectiveness of HIK and other measures for comparing histogram (e.g., the χ^2 distance) have not been extensively compared, the fast computing of Eqn. 1 makes HIK particularly attractive.

In this paper we make three contributions related to HIK SVM learning:

1. We show that HIK is a positive definite (PD) kernel for non-negative real-valued histograms. HIK is known to be a valid kernel for non-negative integer histograms [14]. We give a proof for non-negative real valued histograms. Our proof also completes the missing part of [12], which proved that HIK is conditionally positive definite (CPD).
2. We propose ICD, a fast dual HIK SVM training algorithm. ICD solves the SVM problem without re-encoding the input (which is a necessary step in [12]). It explicitly finds the feature space decision boundary, while the computations are carried out in the input space efficiently. ICD is a deterministic algorithm and do not need to choose a step size for optimization. We empirically show that ICD not only converges faster than the methods of [12,18], but also yields higher accuracies.
3. We show that ICD is robust to the SVM parameter C *in practice*. Choosing SVM parameters by cross validation is very time consuming, but crucial for linear and RBF kernels, or the methods in [12,18]. We empirically show that SVM parameters have only slight effects in ICD thus parameter selection is not necessary.

2 Related Work

The histogram intersection kernel (HIK) is originally proposed by Swain and Ballard for color-based object recognition [17]. It is further shown to be a positive definite kernel when the histograms only contain non-negative integers [14], which makes HIK suitable for SVM classification. HIK is shown to be a conditionally positive definite (CPD) kernel in real-valued cases [12]. We will give a proof that HIK on non-negative real-valued vectors is a positive definite kernel in Sec. 3.1.

HIK has shown to be a suitable similarity measure for comparing two histograms in different machine learning tasks. For example, it achieved higher accuracies in SVM classification than linear or RBF kernel [13,19] in different domains, including object recognition, object detection, place recognition, and scene recognition (whose feature vectors are histograms). In unsupervised learning tasks, kernel k-means clustering using HIK was also shown to produce better visual codebooks (and consequently achieved consistently higher accuracies in the resulting bag of visual words model) than the normal k-means algorithm [19].

A naive method to compute Eqn. 1 will take $O(nd)$ steps, which is very expensive when either n or d is large. However, Eqn. 1 (with different assignment of the weights c_i) is crucial in the training and testing of HIK SVM classifiers, and in HIK based clustering. Recently, [13] showed that Eqn. 1 can be computed

in $O(d \log n)$ steps (and $O(d)$ steps if an approximation is allowed with only slight loss of accuracy). Furthermore, [19] showed that by first quantizing the vectors to integers, exact answer can be obtained in $O(d)$ steps with less overhead than the method in [13]. Fast computation of Eqn. 1 enables the testing of HIK SVM classifiers to have the same complexity as that of linear SVM [13,19], and make HIK clustering almost as fast as the usual k-means clustering [19]. In Sec. 3.2 we will show that the method in [19] is not only a way to accelerate computation, but has a physical interpretation.

These computational methods are also applied in fast training of HIK SVMs, in which Eqn. 1 is again the speed bottleneck. Stochastic gradient descent (SGD) methods are used in [18,12] to train an HIK SVM. More than 10 fold acceleration can be achieved by using the fast method to compute Eqn. 1. PWLSGD [12] is based on the stochastic method Pegasos (Primal Estimated sub-GrAdient SOlver for SVM) [16], and SIKMA [18] is another SGD method. One drawback of these methods is that it is subtle to choose a step size in the gradient descent update, which is important to the success of SGD methods. Also, SGD methods give different results in different runs on the same dataset.

3 The Histogram Intersection Kernel

3.1 HIK in $\mathbb{R}_+$ Is a Positive Definite Kernel

Let $\mathbb{R}_+$ be the set of non-negative real numbers $\{x \geq 0 | x \in \mathbb{R}\}$. We will prove that the histogram intersection kernel $\kappa_{\mathrm{HI}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sum_{j=1}^{d} \min(x_{1,j}, x_{2,j})$ is a valid positive definite kernel for $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}_+^d$.

We use n to denote the number of data points and d for the dimension, and $x_{i,j}$ as the j-th component of a vector $\boldsymbol{x}_i$. We will always use i to index a training example, and use j to index a feature dimension.

We first prove this fact for $d = 1$. Given n real numbers $x_1, \ldots, x_n \in \mathbb{R}_+$, we assume that $x_i \leq x_{i'}$ whenever $1 \leq i < i' \leq n$ without the loss of generality. Thus the kernel matrix K of this set has the property that

$$K_{ii'} = \min(x_i, x_{i'}) = x_{\min(i,i')}. \tag{2}$$

It is easy to verify that $\Lambda = R^T K R$, where R and Λ are defined as:

$$R_{ij} = \begin{cases} 1 & \text{if } i = j \\ -1 & \text{if } i = j - 1 \\ 0 & \text{otherwise} \end{cases}, \quad \text{and} \quad \Lambda_{ij} = \begin{cases} x_1 & \text{if } i = j = 1 \\ x_i - x_{i-1} & \text{if } i = j > 1 \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

The diagonal matrix Λ is positive semidefinite, so is K. Thus κ_{HI} is a positive definite kernel when $d = 1$. The generalization to $d > 1$ is straight forward, because the sum of Mercer kernels is again a Mercer kernel [4].

HIK is proved to be conditionally positive definite (CPD) in $\mathbb{R}$ [12], i.e., $\mathbf{x}^T K \mathbf{x} \geq 0$ when $\sum_j x_j = 0$. However, the proof in [12] is incomplete. The final step of the proof of [12] used the fact that HIK is a positive definite kernel

in $\mathbb{R}_+$, which we have just proved. CPD kernels can be safely used in an SVM if the bias term is included, but may have problem if we do not use the bias term (e.g., the SVM in Eqn 1.1 does not include the bias term.)

One important note is that "HIK is p.d. in $\mathbb{R}_+$" can be proved by setting $\beta = 1$ in Proposition 3 of [1]. Our method, though, provides a new intuitive proof that reveals interesting structures of HIK. It is also worth mentioning that Proposition 3 in [1] can also be easily proved using our technique.

3.2 Equation 1 and Its Feature Space Interpretation

There is a more intuitive way to illustrate that HIK is a Mercer kernel when the histograms only contain non-negative integers [14]. Given a d dimensional histogram $\boldsymbol{x}$, whose elements are all smaller than or equal to an upper bound $\bar{v}$. We define a mapping $B : \mathbb{N} \to \mathbb{R}^{\bar{v}}$ as (i.e., the unary representation)

$$B(x) = [\underbrace{1, 1, \ldots, 1}_{x \text{ times}}, \underbrace{0, 0, \ldots, 0}_{\bar{v}-x \text{ times}}], \tag{4}$$

Then the feature space spanned by κ_{HI} is $d\bar{v}$ dimensional, and a vector $\boldsymbol{x} \in \mathbb{N}^d$ is mapped to $B(\boldsymbol{x}) = [\, B(x_1), \cdots, B(x_d)\,] \in \mathbb{R}^{d\bar{v}}$.

This fact is easy to prove because $xy = \min(x, y)$ when $x, y \in \{0, 1\}$. Thus $\kappa_{\mathrm{HI}}(\boldsymbol{x}, \boldsymbol{y}) = B(\boldsymbol{x})^T B(\boldsymbol{y})$. Using the feature space for integer histograms, we can give a clear interpretation of the method presented in [19] for computing Eqn. 1.

Given a dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ in which we assume that the elements of $\boldsymbol{x}_i$ are non-negative integers not larger than $\bar{v}$, and $y_i \in \{-1, +1\}$. An HIK SVM classifier will be

$$f(\boldsymbol{q}) = \sum\nolimits_{i=1}^{n} \alpha_i y_i \kappa_{\mathrm{HI}}(\boldsymbol{q}, \boldsymbol{x}_i) - \theta \tag{5}$$

for a test example $\boldsymbol{q}$, in which α_i are the Lagrange multipliers. Note that if we set $c_i = \alpha_i y_i$, this equation is a special case of Eqn. 1.

We define a matrix $T \in \mathbb{R}^{d\bar{v}}$ as (in which $c_i = \alpha_i y_i$) $T_{j,k} = \sum_{i:k \geq x_{i,j}} c_i x_{i,j} + k \sum_{i:k<x_{i,j}} c_i$. Then it is shown in [19] that

$$f(\boldsymbol{q}) = \sum\nolimits_{j=1}^{d} T_{j,q_j} - \theta. \tag{6}$$

Now consider the dataset $\{(B(\boldsymbol{x}_i), y_i)\}_{i=1}^n$, i.e., we study the same problem in the feature space instead. Let us assume that a linear SVM in the feature space results in the solution vector $\boldsymbol{w} = [\, \boldsymbol{w}_1, \cdots, \boldsymbol{w}_d] \in \mathbb{R}^{d\bar{v}}$, where $\boldsymbol{w}_i \in \mathbb{R}^{\bar{v}}$ is the weights corresponding to $B(x_i)$. Then we must have

$$f(\boldsymbol{q}) = \boldsymbol{w}^T B(\boldsymbol{q}) - \theta = \sum\nolimits_{j=1}^{d} \boldsymbol{w}_j^T B(q_j) - \theta. \tag{7}$$

Comparing Eqn. 7 and 6, we get that

$$\boldsymbol{w}_j^T B(q_j) = T_{j,q_j} \qquad \forall j \in \{1, \ldots, d\}, q_j \in \{0, 1, \ldots, \bar{v}\}. \tag{8}$$

In other words, we have: for all $j \in \{1, \ldots, d\}$ and $k \in \{0, 1, \ldots, \bar{v}\}$

$$T_{j,k} = \sum\nolimits_{t=1}^{k} w_{j,t}. \tag{9}$$

In short, we just revealed that there is a bijection between the table T and the decision boundary $\boldsymbol{w}$ in the feature space. The key benefit of using the table T is that we do not need to explicitly store $\{B(\boldsymbol{x})\}_{i=1}^{n}$. Also, Eqn. 6 is very efficient ($O(d)$).

3.3 ICD: Intersection Coordinate Descent

This intuition can be used in fast training of HIK SVM classifiers. After quantizing the dataset to integers (with maximum feature value $\bar{v}$), we will solve a linear SVM problem in the feature space $\mathbb{R}^{d\bar{v}}$. However, instead of creating and storing $B(\boldsymbol{x}_i) \in \mathbb{R}^{d\bar{v}}, i = 1, \ldots, n$, we will make use of data structures like the table T to carry out computations in the input space $\mathbb{N}^d$. We do not need to re-encode the input data as the PWLSGD method in [12].

Our method is based on the SVM solver in LIBLINEAR [8], which uses a dual coordinate descent method. Given a dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ where $y_i \in \{-1, +1\}$, its corresponding dataset in the feature space is $\{(B(\boldsymbol{x}_i), y_i)\}_{i=1}^{n}$. A linear SVM in the feature space solves the following problem:

$$\min_{\boldsymbol{w}} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum\nolimits_{i=1}^{n} \xi\left(\boldsymbol{w}; B(\boldsymbol{x}_i), y_i\right), \tag{10}$$

where $\xi\left(\boldsymbol{w}; B(\boldsymbol{x}_i), y_i\right) = \max(1 - y_i\boldsymbol{w}^T B(\boldsymbol{x}_i), 0)^2$ is the L2-loss function. The parameter C controls a trade-off between maximum margin and empirical errors on the training set.

The primal problem (Eqn. 10) is equivalent to the following dual form

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \;\; & g(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T\bar{Q}\boldsymbol{\alpha} - \boldsymbol{e}^T\boldsymbol{\alpha} \\ \text{subject to} \;\; & 0 \le \alpha_i \le U, \forall i, \end{aligned} \tag{11}$$

where $U = \infty$, $\bar{Q} = Q + D$, $Q_{ii'} = y_i y_{i'} B(\boldsymbol{x}_i)^T B(\boldsymbol{x}_{i'})$, D is a diagonal matrix and $D_{ii} = 1/(2C)$ in an L2-loss SVM. Note that $\boldsymbol{\alpha} \in \mathbb{R}^n$ and $\boldsymbol{w} = \sum_i \alpha_i y_i B(\boldsymbol{x}_i)$.

In [8] the dual problem is solved using coordinate descent. The values of α_i are updated sequentially for $i = 1, 2, \ldots, n$. When updating α_i, a new α_i' is chosen such that it will reduce $g(\boldsymbol{\alpha})$ by the largest amount, while still in the range $[0\ U]$. The discriminant function $\boldsymbol{w}$ is then incremented by $(\alpha_i' - \alpha_i) y_i B(\boldsymbol{x}_i)$, i.e., updated using the i-th data point $B(\boldsymbol{x}_i)$. A hypothetical algorithm to solve SVM in the feature space is shown in Algorithm 1.

However, we will not use this algorithm in practice. The main difficulty of applying Algorithm 1 is to compute line 1, 4, and 9 without explicitly constructing the high dimensional vectors $\boldsymbol{w}$ and $B(\boldsymbol{x}_i)$. Instead we will use the table T.

We do not need to compute line 1 because there is a bijection between $\boldsymbol{w}$ and T. A proper initialization of T will replace line 1. We simply initialize all elements of T to 0, which is equivalent to initialize $\boldsymbol{w}$ to 0.

Algorithm 1. A hypothetical algorithm for HIK SVM in the feature space

1: Given $\boldsymbol{\alpha}$ and correspondingly $\boldsymbol{w} = \sum_i \alpha_i y_i B(\boldsymbol{x}_i)$
2: **while** $\boldsymbol{\alpha}$ is not optimal **do**
3: **for** $i = 1, \ldots, n$ **do**
4: $G = y_i \boldsymbol{w}^T B(\boldsymbol{x}_i) - 1 + D_{ii}\alpha_i$
5: $PG = \begin{cases} \min(G, 0) & \text{if } \alpha_i = 0 \\ \max(G, 0) & \text{if } \alpha_i = U \\ G & \text{if } 0 < \alpha_i < U \end{cases}$
6: **if** $|PG| \neq 0$ **then**
7: $\bar{\alpha}_i \leftarrow \alpha_i$
8: $\alpha_i \leftarrow \min(\max(\alpha_i - G/\bar{Q}_{ii}, 0), U)$
9: $\boldsymbol{w} \leftarrow \boldsymbol{w} + (\alpha_i - \bar{\alpha}_i) y_i B(\boldsymbol{x}_i)$
10: **end if**
11: **end for**
12: **end while**

Similarly, Eqn. 6 can be used to efficiently compute $\boldsymbol{w}^T B(\boldsymbol{x}_i)$ in $O(d)$ steps, which makes line 4 easy to compute. The remaining difficulty is then how to update $\boldsymbol{w}$, or equivalently, how to update T because we do not store $\boldsymbol{w}$.

Let us denote $(\alpha_i - \bar{\alpha}_i)y_i$ as δ_{α_i}. Then the change of $\boldsymbol{w}$ (line 9) is now $\Delta\boldsymbol{w} = \delta_{\alpha_i} B(\boldsymbol{x}_i)$, or equivalently ($B(x_{i,j})_t$ is the t-th element of $B(x_{i,j})$),

$$\Delta w_{j,t} = \delta_{\alpha_i} B(x_{i,j})_t, \text{ for all } 1 \leq j \leq d, 1 \leq t \leq \bar{v}.$$

Using Eqn. 9, it is easy to find that

$$\begin{aligned} \Delta T_{j,k} &= \sum_{t=1}^{k} \Delta w_{j,t} \\ &= \sum_{t=1}^{k} \delta_{\alpha_i} B(x_{i,j})_t = \delta_{\alpha_i} \sum_{t=1}^{k} B(x_{i,j})_t \\ &= \delta_{\alpha_i} \min(x_{i,j}, k). \end{aligned} \tag{12}$$

The last equality in Eqn. 12 follows from the identity $\sum_{t=1}^{k} B(x)_t = \min(x, k)$.

In summary, solving an HIK SVM optimization can be done using the dual coordinate descent approach. The computations are carried out on the original histograms instead of in the high dimensional feature space, which maintains the fast training speed. The HIK SVM training algorithm is shown in Algorithm 2, in which we assume that $T_{j,0} = 0$ for any $j \in \{1, \ldots, d\}$. We will refer to Algorithm 2 as the ICD (Intersection Coordinate Descent) method.

After an SVM is trained using ICD, a new example $\boldsymbol{q}$ can be classified in $O(d)$ steps using Eqn. 6, which is the same complexity as that of a linear SVM.

3.4 L1-Loss, Multi-class, Convergence, and All That

The ICD algorithm is provided at `http://www.ntu.edu.sg/home/jxwu` inside the libHIK package. Beyond Algorithm 2, fast HIK SVM training method using

Algorithm 2. ICD: A method for training HIK SVM

{Replace the following lines in Algorithm 1, the remaining lines of Algorithm 1 will be omitted here. }

line 1' : $T_{j,k} \leftarrow 0,$ for all $j \in \{1, \ldots, d\}, k \in \{1, \ldots, \bar{v}\}$

{Note that the below commands update the table T using $\boldsymbol{x}_i$ }

line 4' : $G = y_i \sum_{j=1}^{d} T_{j,x_{i,j}} - 1 + D_{ii}\alpha_i$

line 9' : $T_{j,k} \leftarrow T_{j,k} + (\alpha_i - \bar{\alpha}_i) y_i \min(x_{i,j}, k), \forall j \in \{1, \ldots, d\}, k \in \{1, \ldots, \bar{v}\}$

L1-loss, in primal space, and for multi-class datasets are also provided. Due to the space limit, we will only briefly discuss some important issues.

ICD can use L1-loss function $\xi(\boldsymbol{w}; \boldsymbol{x}_i, y_i) = \max(1 - y_i \boldsymbol{w}^T \boldsymbol{x}_i, 0)$ [8], by setting $U = C$ and $D_{ii} = 0$ in Eqn. 11.

We can accelerate the solving of the primal problem (Eqn. 10) using the same idea of ICD. We maintain both $\boldsymbol{w}$ and the table T during training. There is no need to explicitly create $B(\boldsymbol{x}_i)$. Primal method is preferred when $d \gg n$.

We can solve multi-class problems using the one versus rest method. The Crammer-Singer formulation [3] can also be greatly accelerated by implicit feature space computations using Eqn. 6.

The global convergence Theorem 1 of [8] readily applies to ICD. Thus the ICD method obtains an ϵ-accurate solution in $O(\log(1/\epsilon))$ iterations.

In ICD there are $d\bar{v}$ numbers to change when updating each α_i,[1] while in linear SVM we only need to update d numbers. However, in practice ICD requires a much smaller number of iterations to converge than that of linear SVM. On many real world datasets, ICD converges within 30 iterations, while linear SVM is not converged after 1000 iterations. Thus the training time of ICD is much faster than $\bar{v}$ times of the linear SVM training time. On some difficult datasets ICD is even faster than LIBLINEAR (c.f. Sec. 4).

3.5 Quantization, Default Bin Number, and Default C Parameter

When the feature vectors are not natural integer histograms, we use a simple method to quantize it so that we can apply ICD. Given a dataset, we find $v_{\min}$, the minimum feature value in the training set. We also find $v_{\max}$, which is the 97.5-th percentile of all training feature values.[2] A feature value v is mapped (quantized) to an integer in $[0\ \bar{v}]$ as follows

$$v \rightarrow (\text{int})\,(\bar{v} \times (v - v_{\min})/(v_{\max} - v_{\min}))\,. \tag{13}$$

[1] In practice we do not need to update all these $d\bar{v}$ numbers. If $\bar{v}_j$ is the maximum feature value in dimension j, then we only need to update $T_{j,k}$, $k = 1, \ldots, \bar{v}_j$ for the j-th dimension. We usually observe that $\bar{v}_j \ll \bar{v}$.

[2] We do not use the maximum feature value as $v_{\max}$ because in computer vision it may have an artificial mode at the largest feature value. For example, in the densely sampled bag of visual words model, possibly more than half of the image patches will be mapped to the visual word that corresponds to a uniform image region.

With this simple quantization strategy, we can apply ICD to a much broader range of problems (e.g., whose feature vectors are not natural histograms and have negative feature values).

Choosing $\bar{v}$ is a very important decision. The number of quantization bins, $\bar{v}$, not only affects the training time and storage requirements. It is also directly related to the accuracy of trained classifiers. Obviously a small $\bar{v}$ value will result in low accuracy. However, it is adverse to have a large $\bar{v}$. A large $\bar{v}$ will eventually cause over-fitting and uses more memory and CPU cycles.

We experimented with $\bar{v} = 50$, 100, and 200. In our experiments different problems acquired best accuracies at different $\bar{v}$ values. However, $\bar{v} = 100$ achieved a fair balance between the memory/computation cost and classification accuracy across almost all the datasets. We use $\bar{v} = 100$ if quantization is needed, and if we do not explicitly specify $\bar{v}$ otherwise.

The default value for the C parameter in LIBLINEAR is 1, and feature vectors are usually normalized to the range $[-1\ 1]$. In ICD, κ_{HI} usually generates much large kernel values. Consequently, we choose $C = 10^{-3}$ as the default value for ICD.

4 Experimental Results

We conducted 4 sets of experiments to test various aspects of the ICD algorithm. First we compare ICD with two recently proposed fast HIK SVM training algorithm (Sec. 4.1). We then test ICD on a large scale pedestrian detection dataset (Sec. 4.2). The third set of experiments deal with three different object and scene recognition problems in computer vision (Sec. 4.3). Finally, we show that when cross-validation based SVM parameter selection is infeasible for huge datasets, ICD achieves both faster speed and higher accuracies comparing with linear and RBF kernels, using their default parameter settings (Sec. 4.4). Our empirical results show that ICD is very robust to the C parameter in practice.

Before applying ICD, we use Eqn. 13 to quantize a problem if necessary. We set $\bar{v} = 100$ if not otherwise specified. The one versus rest strategy is used for multi-class problems. We use the default value $C = 10^{-3}$ whenever ICD is used.

4.1 Comparing with PWLSGD and SIKMA

PWLSGD [12] and SIKMA [18] are two recent stochastic gradient descent (SGD) methods for fast training of HIK SVM. In this section we compare ICD with these two, using the software and datasets provided with [12] and [18].

Note that in PWLSGD and SIKMA, we use the SVM parameters that are carefully chosen using cross validation by their corresponding authors. While in ICD we simply use the default value $C = 10^{-3}$. PWLSGD provides sample data on Caltech 101 [7]. We report in Table 1(a) the results when 15 training and testing examples are used in each category. The SIKMA software provides sample data on the PASCAL VOC 2007 images [6]. The comparison results are reported in Table 1(b). SGD methods yield different results on the same dataset

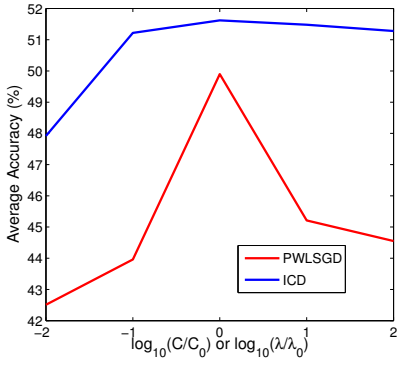
Table 1. Comparing training time and accuracy of HIK SVM methods

ICD		PWLSGD		LIBLINEAR	
56.5	**51.62**%	188.2	49.90%	43.3	48.05%

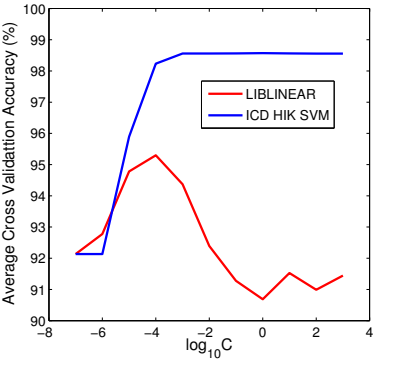
(a) Comparing ICD with PWLSGD

ICD		SIKMA		LIBLINEAR	
9.2	**97.21**%	13.2	96.97±.19%	1.9	96.00%

(b) Comparing ICD with SIKMA



(a) Result on Caltech 101



(b) Result on INRIA

Fig. 1. Effect of different SVM parameters

in multiple runs. Since SIKMA does not fix the seed of its random number generator, we report its average result in 5 runs. Following the setup of SIKMA, we set $\bar{v} = 50$ for this problem.

For every method, we show the training time (in seconds) followed by the classification accuracy. As shown in Table 1(a) and 1(b), ICD not only reduces training time by a large percentage, it also has higher classification accuracies, despite the fact that we do not tune the C parameter in ICD. An additional comparison with LIBLINEAR [8] is provided (with the default settings of LIBLINEAR). The proposed method enjoys higher accuracy with a reasonable amount of increase in training time. For example, on the Caltech 101 dataset, ICD only uses 30% more training time than LIBLINEAR (Table 1(a)).

One attractive property of ICD is that empirically it is not sensitive to SVM parameters. Let $\lambda_0 (= 0.0015)$ and $C_0 (= 0.001)$ denote the SVM parameters used in Table 1 for PWLSGD and ICD respectively. In Fig. 1a we show Caltech 101 results of different C and λ values where $\log_{10}(C/C_0)$ or $\log_{10}(\lambda/\lambda_0)$ ranges from -2 to 2. ICD has high accuracy at the default value C_0, and its accuracy is stable with larger C values. However, large variations are observed for PWLSGD. Time consuming cross-validation based parameter selection is needed for PWLSGD to choose an appropriate λ, but in ICD it is not necessary to choose C. ICD has stable accuracies when $C \geq 10^{-4}$. We observe in Sec. 4.2 again that ICD is robust to C.

Table 2. Results on INRIA pedestrian

	Time	Accuracy	Iterations
ICD	160 s	**98.56**%	21.4
LIBLINEAR	681 s	90.69%	1000

4.2 Pedestrian Detection

Next we use the INRIA pedestrian dataset [5] to evaluate ICD in a large scale vision problem. We use the 256 dimensional CENTRIST [20] visual descriptor as our base feature descriptor. A 108×36 image patch is divided into 9×4 blocks. Any neighboring 2×2 blocks are formed into a super-block. The concatenation of CENTRIST in all super-blocks generates a feature vector that has 6,144 dimensions. This feature vector is a natural histogram with $\bar{v} = 352$. We evaluate the SVM training algorithms in a "hard" dataset that is the result of bootstrapping the INRIA dataset (using the procedures in [5]). There are 30,711 examples. This dataset is mostly dense, resulting in a large scale problem with approximately 82 million non-zero feature values.

We compare ICD with LIBLINEAR. Five-fold cross validation is applied. The total training time, average accuracy, and average number of iterations needed to finish the optimization are reported in Table 2. Default C values are used in both methods in Table 2.

One interesting observation from Table 2 is that ICD only takes about 24% of the training time of LIBLINEAR. This is related to the number of iterations which is required to terminate the SVM optimization. HIK SVM has higher discrimination capability than a linear SVM, and ICD usually requires a small number of iterations to converge in practice.[3] In summary, ICD scales well to large problems, and is particularly suitable when the feature vectors are natural histograms.

The effect of SVM parameters are studies in Fig 1b, where the C value ranges from 10^{-7} to 10^{3}, with step size 10. Linear SVM is sensitive to C, and an overly large C will lead to a lower accuracy. In this dataset HIK SVM is not sensitive to C: the accuracy increases with C and a large C value will not lead to a lower accuracy. The same phenomenon is observed in almost all datasets we tested in this paper.

It is also interesting to compare with general purpose SVM learners with linear, RBF, or the histogram intersection kernel. However, on this large scale dataset, general purpose SVM solvers (e.g., LIBSVM [2]) requires more than 10 hours to converge. This fact makes these solvers impractical for large problems.

4.3 Object and Scene Recognition

In this section we evaluate ICD in 3 more benchmark vision problems: Caltech 101 [7], 15 class scene recognition [9], and 8 class sport events [10]. Images

[3] LIBLINEAR terminates when the iteration number is 1000. So the actual iterations needed for convergence is higher than 1000 in this problem.

Table 3. Results on various vision problems

	caltech			scene			sports		
	Time	Acc	Acc(cv)	Time	Acc	Acc(cv)	Time	Acc	Acc(cv)
ICD	71.5	60.0%	59.9%	20.1	81.9%	82.0%	10.4	81.3%	81.5%
LIBSVM+HIK	66.8	54.6%	54.9%	40.1	81.6%	81.7%	10.7	81.0%	81.0%
LIBLINEAR	6.3	54.1%	54.2%	1.4	75.3%	75.5%	0.5	76.7%	78.5%
LIBSVM+LIN	65.2	51.3%	51.4%	33.5	76.8%	76.8%	8.5	78.8%	78.8%
LIBSVM+RBF	67.3	18.2%	53.4%	58.2	35.2%	79.6%	14.1	12.5%	76.5%

are represented using the bag of visual words model, and feature vectors are generated by libHIK [19] with k-means visual codebooks. The results from the first train/test split of libHIK are reported.[4]

Three kernel types (linear, RBF, and HIK) are compared. The features are quantized for ICD and LIBSVM+HIK, because we want these two methods to use exactly the same data. In Table 3, the first two columns for each dataset report the training time and accuracy of a method when we use the default SVM parameters. We also use training set cross validation to choose SVM parameters in the range $\log_{10} C \in [-5 \;\; 3], \log_{10} \gamma \in [-5 \;\; -1]$, whose accuracies are reported in the 'Acc(cv)' column.

In terms of classification accuracy, HIK has clear advantages over linear and RBF kernel types. ICD achieves slightly higher accuracies than the general purpose LIBSVM solver with HIK. The Caltech 101 dataset shows an exception where ICD has a large 5.4% advantage over LIBSVM. This might be due to the fact that there are 101 classes in this problem, while the one versus one strategy of LIBSVM is not suitable for handling large number of classes.

In terms of training time, LIBLINEAR trains much faster than other methods, while all other methods have comparable training time. It is worth noting that the feature vectors are $d = 6,200$ dimensional in these problems, while the training set size n ranges from 560 to 1515. Dual space algorithms (including the proposed method) is not effective while $d \gg n$. However, ICD still has approximately the same training speed as LIBSVM.

Again we observed the phenomenon that ICD is not sensitive to SVM parameters, since 'Acc(cv)' only has slight advantage over 'Acc' (the accuracy using default ICD SVM parameters). The robustness to SVM parameters is very attractive because cross validation parameter selection is infeasible for huge datasets, e.g., accuracy of the RBF kernel is heavily affected by SVM and kernel parameters.

4.4 Working with Non-histograms and Default SVM Parameters

Nowadays many problems are too large to perform cross validation for SVM parameter selection. Thus it is important to emphasize *high accuracies using the default SVM parameters*. In the final set of experiments, we will evaluate ICD

[4] Note that the Caltech 101 features are different than those used in Sec. 4.1.

Table 4. Properties of the non-histogram datasets

	train size	test size	dimension	#class
ijcnn1	4,990	91,701	22	2
shuttle	43,500	14,500	9	7
acoustic	78,823	19,705	100	3
rcv1	677,399	20,242	47,236	2

Table 5. Comparing performances using default SVM parameters

	ijcnn1		shuttle		acoustic		rcv1	
	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy
ICD	0.6	94.82%	0.7	99.50%	137.2	82.98%	34.3	97.95%
LIBSVM+HIK	42.9	95.27%	3.7	99.57%	1362.0	83.12%	$> 50,000$	
LIBLINEAR	0.4	91.79%	0.8	92.35%	14.6	80.18%	6.4	97.96%
LIBSVM+LIN	48.4	92.12%	7.4	97.10%	1736.0	80.46%	$> 50,000$	
LIBSVM+RBF	60.1	92.79%	14.3	97.23%	1824.0	79.69%	$> 50,000$	

on problems that are not natural histograms and on huge datasets. The four problems we experimented with are chosen from the LIBSVM dataset collection: ijcnn1, shuttle, acoustic (combined), and rcv1 (binary). We choose these datasets whose features are not histograms, and whose sizes range from medium to huge. The particulars of these problems are collected in Table 4. We switched the training and testing set of the rcv1 problem so that we have a huge training set to test the scalability of ICD.

We compare with LIBLINEAR and LIBSVM using linear, RBF, and HIK. We use the default value $C = 1$ for LIBLINEAR and LIBSVM on the original feature vectors, and use $C = 10^{-3}$ on quantized versions. Experimental results are reported in Table 5.

One important observation is that on these datasets both HIK SVM classifiers (ICD and LIBSVM+HIK) achieve higher accuracies even when their feature vectors are not natural histograms. We also compare the two pairs of methods that use the same kernel. Although the LIBSVM+LIN solver sometimes have noticeable advantage over LIBLINEAR at the cost of much longer training time (e.g., in the shuttle problem), the proposed ICD method has almost the same accuracy as LIBSVM+HIK.

ICD, however, trains a lot faster than LIBSVM+HIK, and its speedup is related to size of the datasets. ICD is about 5 times faster than LIBSVM+HIK on the shuttle dataset. However, in the rcv1 dataset, the speedup is more than 3 orders of magnitude. For all three kernel types, the LIBSVM solver did not converge after 50,000 seconds when running the rcv1 problem. Thus LIBSVM's accuracies on this dataset is not available in Table 5.

LIBLINEAR is faster than ICD. However, the speedup is usually smaller than 10. Given the fact that the training time is smaller than 1 minute even in the rcv1 dataset, we believe that the proposed algorithm is preferable for its higher classification accuracies.

It is generally accepted that for problems with a large number of feature dimensions, linear SVM usually works as well as other more complex kernel types. Thus it is not surprising to observe that on the rcv1 dataset, ICD requires more training time than LIBLINEAR, while both methods have approximately the same classification accuracy. Our experiments on rcv1, though, further illustrate the scalability of ICD. In computer vision, we usually work with a medium dimensional feature vector (e.g., around 5000, smaller than that of rcv1). The experiments on rcv1 illustrate that ICD is able to handle even millions of training examples in computer vision problems.

5 Conclusions and Future Work

Our contributions are threefold. First, we prove that the histogram intersection kernel (HIK) is a positive definite kernel for non-negative real numbers. Second, we give the physical meaning of the computational method that accelerates the kernel evaluation of HIK. Based on this interpretation, we propose ICD, a fast, accurate, and scalable HIK SVM solver. Third, we empirically show that ICD is not sensitive to the C parameter in SVM, and achieve high accuracies using its default settings on huge datasets.

As a summary of the theoretical analyses and experimental results, we list the advantages and limitations of the proposed method (+ for advantages and - for limitations).

Speed (+). ICD trains much faster than general purpose SVM solvers. It also trains faster than two recent SGD based methods (PWLSGD and SIKMA). The testing speed has the same complexity as linear classifiers.

Insensitivity to C (+). Accuracy of ICD generally increases with the SVM parameter C. However, a large C does not lead to low accuracy. This empirical property is particularly attractive on huge datasets where cross validation parameter selection is infeasible.

Scalability (+). It scales easily to large problems, and is most efficient for problems with a medium number of feature dimensions and a huge number of training examples. Many vision problems fit into this category.

Accuracy (+). It has comparable accuracies to general purpose SVM solvers, and the PWLSGD or SIKMA HIK SVM solver.

Simplicity (+). There is only 1 parameter in ICD (C), and the default value $C = 10^{-3}$ works well in most problems. It is also a deterministic algorithm, producing the same result on multiple runs. There is no need to re-encode input data.

Storage (-). The table T increases the storage cost, especially when d is large, and when the problem contains a large number of classes.

Quantization (-). The need for quantization introduces additional costs (although this cost is small), and the quantized version does not guarantee higher accuracy than linear SVM in a few cases (e.g., rcv1).

There are possible directions to address certain limitations of the proposed method. For example, we can make the table T sparse, which will answer the

storage problem at the cost of a reasonable increase in training and testing time. A rule of thumb can be developed to automatically choose between a linear SVM or HIK SVM. Finally, we can explore adaptive quantization methods to achieve better quantized feature vectors (and higher accuracies).

References

1. Boughorbel, S., Tarel, J.P., Boujemaa, N.: Generalized histogram intersection kernel for image recognition. In: ICIP (2005)
2. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm
3. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. JMLR 2, 265–292 (2001)
4. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, Cambridge (2000)
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, vol. 1, pp. 886–893 (2005)
6. Everingham, M., Gool, L.V., Williams, C., Winn, J., Zisserman, A.: The PASCAL visual object classes challenge 2007 (VOC 2007) results. Tech. rep (2007)
7. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training example: an incremental bayesian approach tested on 101 object categories. In: CVPR 2004, Workshop on Generative-Model Based Vision (2004)
8. Hsieh, C.J., Chang, K.W., Lin, C.J., Keerthi, S.S., Sundararajan, S.: A dual coordinate descent method for large-scale linear SVM. In: ICML, pp. 408–415 (2008)
9. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR, vol. II, pp. 2169–2178 (2006)
10. Li, L.J., Fei-Fei, L.: What, where and who? Classifying events by scene and object recognition. In: ICCV (2007)
11. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)
12. Maji, S., Berg, A.C.: Max-margin additive classifiers for detection. In: ICCV (2009)
13. Maji, S., Berg, A.C., Malik, J.: Classification using intersection kernel support vector machines is efficient. In: CVPR (2008)
14. Odone, F., Barla, A., Verri, A.: Building kernels from binary strings for image matching. IEEE Trans. Image Processing 14(2), 169–180 (2005)
15. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE TPAMI 24(7), 971–987 (2002)
16. Shalev-Shwartz, S., Singer, Y., Srebro, N.: Pegasos: Primal estimated sub-gradient solver for svm. In: ICML, pp. 807–817 (2007)
17. Swain, M.J., Ballard, D.H.: Color indexing. IJCV 7(1), 11–32 (1991)
18. Wang, G., Hoiem, D., Forsyth, D.: Learning image similarity from flickr groups using stochastic intersection kernel machines. In: ICCV (2009)
19. Wu, J., Rehg, J.M.: Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel. In: ICCV (2009)
20. Wu, J., Rehg, J.M.: CENTRIST: A visual descriptor for scene categorization. Tech. Rep. GIT-GVU-09-05, GVU Center, Georgia Institute of Technology (2009)

Weakly-Paired Maximum Covariance Analysis for Multimodal Dimensionality Reduction and Transfer Learning

Christoph H. Lampert[1] and Oliver Krömer[2]

[1] Institute of Science and Technology Austria, Klosterneuburg, Austria
[2] Max Planck Institute for Biological Cybernetics, Tübingen, Germany

Abstract. We study the problem of multimodal dimensionality reduction assuming that data samples can be missing at training time, and not all data modalities may be present at application time. *Maximum covariance analysis*, as a generalization of PCA, has many desirable properties, but its application to practical problems is limited by its need for perfectly paired data. We overcome this limitation by a latent variable approach that allows working with weakly paired data and is still able to efficiently process large datasets using standard numerical routines. The resulting *weakly paired maximum covariance analysis* often finds better representations than alternative methods, as we show in two exemplary tasks: texture discrimination and transfer learning.

1 Introduction

With the increasing availability of cheaper sensors, multimodal data has become nearly ubiquitous in practical computer vision tasks: images on the web have text captions, videos have audio tracks, and modern mobile phones can even record acceleration data in addition to their audio and visual recording capabilities. However, the field of multimodal data processing so far plays only a minor role in current computer vision research, where most algorithms are only able to process one data domain at a time. Those multimodal algorithms that do exist typically make restrictive assumptions, such as a priori known pairings between all data samples. They also commonly require that all sensor information is available reliably at all times, which is not always the case in practical problems because the use of multiple sensors increases the risk of subsystems failing.

In this paper, we introduce a dimensionality reduction method that can handle weakly paired data, and that is robust again the risk of partially missing data. Furthermore it incorporates two further advantages, which are of great importance for practical applications: it is simple, and it is efficient. By simplicity we mean that the method is based on elementary principles, in our case derived from statistics, which can be easily implemented and understood by a outsider of the field. An efficient method can be applied to data sets of realistic size, i.e. at least several thousand data vectors with thousands of dimensions.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 566–579, 2010.

2 Multimodal Dimensionality Reduction

We assume that we are given related data samples in two or more data modalities of potentially very high dimension. The general goal of *multimodal dimensionality reduction* is to compute new representations for these data samples that lie in lower-dimensional feature spaces. In comparison to normal, unimodal, dimensionality reduction, we expect the availability of multiple data representations to give a better indication of what the true signal in the data is, that we want to retain, and what parts are noise that can be suppressed. As motivated in the introduction, we are interested in robust techniques that can handle missing examples in the original data. Additionally, once good dimensionality reduction mappings have been found, we want to be able to process each modality separately, in order to handle situations wherein some modalities are not always accessible. We formalize these intuitions in the following definitions.

Definition 1 (Inductive Dimensionality Reduction). *Let* $X = (x_1, \ldots, x_n) \subset \mathbb{R}^{d \times n}$ *be a set of data vectors. We call a procedure* ***inductive dimensionality reduction*** *if, given the input* X*, it outputs a functional mapping* $f : \mathbb{R}^d \to \mathbb{R}^q$ *with* $q < d$*. The image of* X *under* f *we call a* ***lower-dimensional representation*** *of* X *and denote it by* $\hat{X} = (\hat{x}_1, \ldots, \hat{x}_n)$*, i.e.* $\hat{x}_i = f(x_i)$*.*

In the rest of this paper, we will only consider inductive methods, which include *PCA* [25], *kernelPCA* [28] and *autoencoder networks* [10]. Non-inductive methods, e.g. *probabilistic latent semantic analysis (pLSA)* [11], and *Isomap* [30], also compute a lower-dimensional representation $\hat{X}$ from X, but do not provide a function f that could be applied to future data.

The two main families of inductive dimensionality reduction techniques, discriminative and generative, differ in the applications they are suitable for: discriminative techniques, such as *linear discriminant analysis (LDA)* [6] and *canonical correlation analysis (CCA)* [2,12], identify lower-dimensional representations that are suitable for a specific task that has to be known at the time of data processing, e.g. classification into a known set of classes. By discarding all signal dimensions that are not relevant for the specified task, discriminative techniques can often achieve a large reduction in dimensionality without loss of accuracy. Their drawback is that the representations found might not be well suited to tasks different from the specified one. In this work we concentrate on generative dimensionality reduction instead, where the goal is to find lower-dimensional data representations that are suited for various subsequent tasks, not just for a specific one. Intuitively, generative dimensionality reduction techniques can be seen as data compression methods, because it is often possible to recover the original data from the reduced representation with usually only a small reconstruction error.

Definition 2 (Multimodal Dimensionality Reduction) *Let* $X^{(1)} = (x_1^{(1)}, \ldots, x_{n_1}^{(1)}) \subset \mathbb{R}^{d_1 \times n_1}, \ldots, X^{(m)} = (x_1^{(m)}, \ldots, x_{n_m}^{(m)}) \subset \mathbb{R}^{d_m \times n_m}$ *be several data sets from potentially different spaces. We call an inductive dimensionality reduction technique* ***multimodal*** *if, given the inputs* $X^{(1)}, \ldots, X^{(m)}$*, it outputs functions* $f_1 : \mathbb{R}^{d_1} \to \mathbb{R}^q, \ldots, f_m : \mathbb{R}^{d_m} \to \mathbb{R}^q$ *for all data domains.*

Clearly, every inductive dimensionality reduction technique can in principle be used in a multimodal framework by just processing each data domain independently. However, since in the multimodal setup the functions f_i can depend on all data sets and not just on $X^{(i)}$ itself, one would expect multimodal techniques to use this information to find better representations than those of unimodal methods. The canonical way to construct multimodal algorithms is by making use of dependencies between the data samples that are induced by *pairings*:

Definition 3 (Weakly Paired Multimodal Data). *We call a collection of data sets $X^{(1)}, \ldots, X^{(m)}$ **weakly paired**, if each $X^{(i)}$ is split into k groups as*

$$X^{(i)} = (x^{(i)}_{1,1}, \ldots, x^{(i)}_{1,n^i_1}, \ldots, x^{(i)}_{k,1}, \ldots, x^{(i)}_{k,n^i_k}) \in \mathbb{R}^{d_i \times n_i} \tag{1}$$

*with $n_i = \sum_{l=1}^{k} n^i_l$. The special cases where $n^i_l = 1$ for all $i = 1, \ldots, m$ and $l = 1, \ldots, k$ we call **fully paired**. The other extremal case is $k = 1$, which we call the **unpaired** situation.*

Weakly paired data is common in multimodal data processing. For example, in video processing the groups could correspond to separate scenes for which we have data in the modalities: visual content, audio soundtrack, and textual subtitles. Unfortunately, existing techniques require fully paired data, which can introduce artificially overconstrained systems. In the above video example, one could pair each frame with the audio and subtitle content shown simultaneously with it. However, many of the correspondences introduced this way will be incorrect, as the synchronization between visual and other content is typically on a time scale much larger than the individual frame label.

3 Weakly Paired Maximum Covariance Analysis

In this section we derive a method for inductive multimodal dimensionality reduction with weakly paired data that we call *weakly paired maximum covariance analysis (WMCA)*. It can handle weakly paired and even unpaired data, because it infers suitable pairings directly from the data instead of requiring them a priori. This makes WMCA robust against missing data and enables it to process datasets where the domains have different numbers of samples, whereas previous techniques only worked if $n_1 = \cdots = n_m$ and the data was fully paired.

3.1 Linear Weakly Paired Covariance Maximization

We first study linear multimodal dimensionality reduction, and in order to simplify the notation we restrict the discussion to two modalities $X \in \mathbb{R}^{d \times n}$ and $X' \in \mathbb{R}^{d' \times n'}$. We will discuss the non-linear case in Section 3.2, and the extension to more than two modalities in Section 3.3.

In linear dimensionality reduction the dimensionality reduction functions can be written as $f(x) = W^t x$ for a matrix $W \in \mathbb{R}^{d \times q}$, and $f'(x') = W'^t x'$ for a matrix $W' \in \mathbb{R}^{d' \times q'}$. The lower dimensional representations are thus $\hat{X} = W^t X$

and $\hat{X}' = W'^t X'$. Typically, W and W' are assumed orthogonal matrices, so they contain the basis vectors of the linear subspaces of $\mathbb{R}^d$ and $\mathbb{R}^{d'}$ to be retained.

The most popular technique for generative linear dimensionality reduction is *principal component analysis (PCA)*. PCA finds a lower-dimensional representation that retains as much of the original signal's variance as possible. PCA can also be used to process fully paired multimodal data (by stacking the data vectors), but this does not qualify as a multimodal technique in the sense of Definition 2, since the construction requires that all modalities are also present in future data. The truly multimodal counterpart to PCA is *maximum covariance analysis (MCA)* [31], which would be ideal for our purposes, except that it also requires fully paired data.

Definition 4 (Maximum Covariance Analysis). *Let X and X' be fully paired datasets, i.e. for $X = (x_1, \ldots, x_n)$ and $X' = (x'_1, \ldots, x'_n)$ there is a pairing between each x_i and x'_i. Let X and X' be* centered*, i.e. $\frac{1}{n}\sum_{i=1}^n x_i = 0$ and $\frac{1}{n}\sum_{i=1}^n x'_i = 0$. **Maximum covariance analysis (MCA)** performs multimodal dimensionality reduction with projection matrices W, W' that solve*

$$\max_{W,W'} \operatorname{tr}\left[W^t X X'^t W'\right] \tag{2}$$

where the maximization runs over all orthogonal $d \times q$ and $d' \times q$ matrices.

Note that the condition of centered data is not severe, as we can center every dataset by subtracting the data mean from all samples.

MCA gets its name from the fact that the objective function (2) measures the total covariance between the individual dimensions of $\hat{X} = W^t X$ and $\hat{X}' = W'^t X'$, as one can see from rewriting $\operatorname{tr}[W^t X X'^t W'] = \sum_{p=1}^q [W^t X]_p^t [W'^t X']_p$ where $[.]_p$ indicates the p-th column.

Even though MCA is a strong method for multimodal dimensionality reduction, it has found relatively little application in computer vision contexts. We believe that the main reason for this is that MCA requires fully paired data, which realistic computer vision tasks often do not provide. In the rest of this section, we show how MCA can be extended to the weakly paired situation, calling the result *weakly paired maximum covariance analysis (WMCA)*.

Definition 5 (Weakly Paired Maximum Covariance Analysis). *Let X and X' be centered data sets that are weakly paired as specified in Definition 3. **Weakly paired maximum covariance analysis (WMCA)** performs multimodal dimensionality reduction with projection matrices W and W' that solve*

$$\max_{W,W',\Pi} \operatorname{tr}\left[W^t X \Pi X'^t W'\right], \tag{3}$$

where W and W' run over all orthogonal $d \times q$ matrices and $d' \times q$ matrices, respectively. Π runs over all $n \times n'$ pairing matrices *that respect the group structure of X and X', i.e. $\Pi = \operatorname{diag}(\Pi^1, \ldots, \Pi^k)$, where for $l = 1, \ldots, k$ we have $\Pi^l \in \{0,1\}^{n_l \times n'_l}$ such that $\sum_{i=1}^{n_l} \Pi^l_{i,j} \leq 1$ for all $j = 1, \ldots, n'_l$ and $\sum_{j=1}^{n'_l} \Pi^l_{i,j} \leq 1$ for all $i = 1, \ldots, n_l$.*

There is no single closed form solution to the optimization (3), as it requires both continuous optimization for W and W', and combinatoric optimization for Π. Furthermore, it is a high-dimensional non-convex problem, such that finding the global optimum in a numeric procedure is typically not possible. We can, however, efficiently find a locally optimal solution by *alternating maximization*:

- For known Π, solve

$$W, W' = \text{argmax}_{W,W'} \ \text{tr}\left[W^t X \Pi X'^t W'\right] \tag{4}$$

  Because Π is assumed to be known, the structure of this maximization is the same as when performing MCA with fully paired data. We obtain the basis vectors that form W and W' by computing the SVD of the matrix $X\Pi X'^t \in \mathbb{R}^{d \times d'}$, and keeping the q components in both domains with the largest singular values. When q is much smaller than d and d' (which is the typical case), we can use techniques for accelerated SVD computation, e.g. based on random projections [24]. This allows the efficient solution of Equation (4) even when d and d' are in the range of thousands or larger.
- For known W and W', solve

$$\Pi = \text{argmax}_{\Pi} \ \text{tr}\left[W^t X \Pi X'^t W'\right]. \tag{5}$$

  Given that $\text{tr}\left[W^t X \Pi X'^t W'\right] = \text{tr}\left[X'^t W' W^t X \Pi\right]$ and Π's special properties, the optimization (5) corresponds to a *linear assignment problem* with cost matrix $[X'^t W' W^t X]^t \in \mathbb{R}^{n \times n'}$. Furthermore, because of the diagonal block structure of Π, we can solve k separate problems of size $n_k \times n'_k$ instead of one big one of size $n \times n'$. Consequently, Equation (5) remains solvable in an efficient way even for large sample sizes, e.g. using the Hungarian algorithm [14] or LAPJV [13].

In both steps of the algorithm we maximize the same objective function. Therefore its value will increase monotonically over the iterations, which provides us with a natural stop criterion; we have reached a local maximum if the objective value does not increase any further.

To obtain a complete algorithm, we need a start value for Π. Unless some reasonable pairing is known a priori, we use $\Pi = \text{diag}(\Pi^1, \ldots, \Pi^k)$ with $\Pi^k \equiv \frac{1}{n_k n'_k}$. This is not a pairing matrix in the sense defined above, but it ensures that all data samples have influence on the initial choice of W and W'. The pairing property of Π will be established during the first solution of the maximization (5). As the alternating optimization algorithm is only locally convergent, it could also be run multiple times from different, e.g. random, start configurations. In our experiments, this did not lead to noticeable improvement, indicating that the above choice of Π is already a good heuristic.

3.2 Nonlinear Weakly Paired Covariance Maximization

Nonlinear dimensionality reduction techniques are often more powerful than linear ones, because they have more flexibility in the dimensionality reduction function that they output. MCA and WMCA can be made into non-linear techniques

by *kernelization*. As the necessary steps are very similar to, e.g., the derivation of kernelPCA from PCA we only outline them here, and refer the reader to [28] for a more detailed description of kernelization.

For kernelization, we require positive definite and symmetric similarity measures between samples, called kernel functions, that we denote by $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ and $k' : \mathbb{R}^{d'} \times \mathbb{R}^{d'} \to \mathbb{R}$. Arguments from functional analysis show that any such kernel function corresponds to an inner product in a latent Hilbert space, and that it induces a latent feature map from the original data domain to this space [28]. Kernelized WMCA now consists of mapping the input data into the latent Hilbert spaces and performing linear WMCA on the resulting data sets. In the kernelized form, the optimization problem (3) becomes

$$\max_{A,A',\Pi} \ \operatorname{tr}\left[A\bar{K}\Pi\bar{K}'A'^t\right], \tag{6}$$

where $\bar{K}$ and $\bar{K}'$ are the centered kernel matrices. $\bar{K}$ is computed by forming the kernel matrix $K \in \mathbb{R}^{n\times n}$ as $[K]_{ij} = k(x_i, x_j)$ and then centering it using the formula $\bar{K} = K - \frac{1}{n}\mathbf{1}_n K - \frac{1}{n}K\mathbf{1}_n + \frac{1}{n^2}\mathbf{1}_n K\mathbf{1}_n$, where $\mathbf{1}_n$ denotes the $n \times n$ matrix in which all elements are 1. $\bar{K}'$ is computed from k' in the analogous way. Centering the kernels ensures that the implicitly defined feature vectors have zero mean in the latent feature space.

We solve the optimization problem (6) with the same alternating optimization scheme described previously with two differences:

- In contrast to W, W', the matrices $A \in \mathbb{R}^{n\times q}$ and $A' \in \mathbb{R}^{n'\times q}$ are not orthogonal. Instead they have to fulfill conditions $A^tKA = \mathrm{Id}$ and $A'^tK'A' = \mathrm{Id}$, which expresses orthogonality in the latent feature space. We obtain the rows of A and A' from a generalized eigenvalue problem:

$$\begin{pmatrix} 0 & K\Pi K' \\ K'\Pi^t K & 0 \end{pmatrix} \begin{pmatrix} a \\ a' \end{pmatrix} = \lambda \begin{pmatrix} K & 0 \\ 0 & K' \end{pmatrix} \begin{pmatrix} a \\ a' \end{pmatrix}. \tag{7}$$

  Solving Equation (7) is computationally more costly than solving (4). However, because we are interested only in the q eigenvectors of highest eigenvalue, we can still solve it efficiently using, e.g., the *power method* [7].
- When solving for A and A' in this way, the matrix $K\Pi K$ is of size $n \times n'$ instead of $d \times d'$. In the case where the number of data samples is smaller than the number of original data dimensions, it can be advantageous to use the kernelized formulation (6) also for the linear case. For this, one uses linear kernels $k(x, \tilde{x}) = x^t\tilde{x}$ and $k'(x', \tilde{x}') = x'^t\tilde{x}'$ and obtains the solutions of the problem (4) as $W = A^tX$ and $W' = A'^tX'$.

Kernelized WMCA provides reduction functions $f : \mathbb{R}^d \to \mathbb{R}^q$ and $f' : \mathbb{R}^{d'} \to \mathbb{R}^q$ by setting $f(x) = A^tK(x)$ with $K(x) = (k(x, x_1), \ldots, k(x, x_n))^t$ and $f'(x') = A'^tK'(x')$ with $K'(x') = (k'(x', x'_1), \ldots, k'(x', x_{n'}))^t$. Thus it is an inductive multimodal dimensionality reduction technique. Besides its flexibility to learn nonlinear projection functions, kernelization has another advantage. It allows us to process data sources that are provided in a different form than as vectors, e.g. text documents or graphs. In such scenarios, only a similarity measure, with the properties of a kernel function, needs to be defined to create Equation (6).

3.3 WMCA for More than Two Modalities

So far, we described WMCA for two data sources. An extension to more than two modalities is straightforward by reformulating the objective function as the sum of all pairwise covariances between all modalities. Thus, Equation (3) becomes

$$\max_{\substack{W^{(1)},\dots,W^{(m)} \\ \Pi^{(1,2)},\dots,\Pi^{(m-1,m)}}} \operatorname{tr}\Big[\sum_{i,j=1}^{m} W^{(i)} X^{(i)t} \Pi^{(i,j)} X^{(j)} W^{(j)t}\Big], \tag{8}$$

with the convention that $\Pi^{(i,i)} = 0$ and $\Pi^{(i,j)} = \Pi^{(j,i)t}$, and Equation (6) into

$$\max_{\substack{A^{(1)},\dots,A^{(m)} \\ \Pi^{(1,2)},\dots,\Pi^{(m-1,m)}}} \operatorname{tr}\Big[\sum_{i,j=1}^{m} A^{(i)} \bar{K}^{(i)t} \Pi^{(i,j)} \bar{K}^{(j)} A^{(j)t}\Big]. \tag{9}$$

Both systems can be solved by alternating maximization, where the step of finding the projection directions is solvable as an eigenvalue problem (generalized for the kernelized case), and finding the sample pairings requires solving $\frac{1}{2}m(m-1)$ linear assignment problems. Note that this quadratic scaling in the number of modalities does not pose a practical problems, since the majority of multimodal datasets utilize only a small number of modalities.

4 Related Work

As a classical dimensionality reduction technique, MCA comes from the same family of standard statistical methods as PCA, LDA and CCA. It also forms the basis for *partial least squares (PLS)* regression (PLS) [33]. Over the last 10 years, all of these techniques have been kernelized into non-linear versions [3,27,28]. The kernelization approach we take in Section 3.2 is similar to these, and the resulting expressions resemble the ones for *kernel canonical correlation analysis (kernelCCA)* [9]. KernelCCA also acts on multimodal data, but it would not have been a suitable basis for our purposes, as it is not generative. Furthermore, kernelCCA requires a priori setting of a regularization parameter for each modality, whereas, except for the number of output dimensions, MCA and WMCA are parameter-free. Nevertheless, CCA and kernelCCA are probably the most common methods for multimodal dimensionality reduction, typically in situations with a single fixed target application, e.g. *fMRI analysis* [8], *image clustering* [5], *speaker identification* [18], or *shape recovery* [16]. Alternative approaches include *multimodal pLSA* [17] or *Hilbert-Schmidt dependence maximization* [4], but these require a more careful experimental setup and are computationally more demanding. In contrast, the classical methods, and also WMCA, can be implemented with off-the-shelf components, typically just matrix operations.

To our knowledge, WMCA is the first multimodal dimensionality reduction technique that can efficiently handle weakly-paired data in the sense of Definition 3. The idea of treating unknown correspondences as latent variables and

optimizing over them, however, has been used in previous applications, including the classical k-means [20] algorithm, where one alternates between the centroid computation and the cluster assignment. An optimization similar to ours occurs in [4], which also alternates between a search for projection directions and for assignments. However in both cases the assignments are between sample and clusters, not between samples in different data modalities. WMCA's aspect of identifying relevant elements in groups of samples is somewhat related to witness approaches in *multiple instance learning* [1]. However, the criterion by which the elements are identified and the overall problem framework are very different.

5 Experimental Evaluation

In this section we show that due to its use of multimodal information, WMCA is often able to find low dimensional representations that reflect the information content of a data source better than a unimodal treatment of the same data. For this, we perform experiments on two realistic datasets: one for texture discrimination and one for transfer learning.

5.1 Texture Discrimination

As described in the introduction, generative dimensionality reduction aims at finding data representations that are suitable for different subsequent tasks. In this section we study this by performing texture discrimination both as an unsupervised and as a supervised learning problem. Note that both scenarios occur in real world scenarios. For example, in robot navigation it is important to classify surfaces into a set of known classes, such as *road* or *quick sand* (supervised). However, in order to collect probes in a new environment, the robot also needs to be capable of handling previously unobserved surface types, e.g. by grouping them based on their material properties (unsupervised).

To perform experiments on both setups we use a multimodal *Materials* dataset[1] that consists of images as well as audio signatures for 17 different materials (e.g. *bricks*, *styrofoam*, *wallpaper*, and *woven carpet*), see Figure 1. In contrast to available datasets with artificially constructed perfect pairings, the situation for this data is closer to the real problems that occur in multimodal data acquisition. The audio signal is recorded by dragging a small audio probe over the textured surfaces multiple times, and measuring the induced characteristic vibrations with a microphone. The images are captured using an ordinary digital camera. It is a priori unknown how a meaningful pairing should be constructed between the audio signals, which reflect a trajectory over the surface, and the rectangular regions depicted in the images. Also the conditions under which both modalities can be obtained differ: to capture images, one needs acceptable viewing conditions (e.g. no dust or fog). However, once this situation is established, each image contains a large amount of information from different physical locations. Audio recording in the described setup works by physical contact to

[1] The data set and source code are available at `http://www.ist.ac.at/~chl`

the material. The sensor can be shielded from environmental influences, but the information obtained is only very local.

We demonstrate how multimodal dimensionality reduction can be beneficial under such conditions by adopting an asymmetric multimodal setup: we use image and audio data to compute dimensionality reduction function, but we assume that only audio information is available at the time of application.

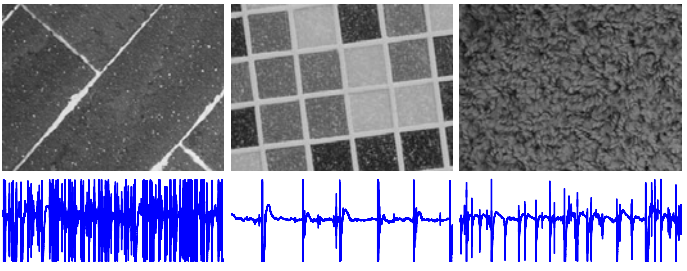


Fig. 1. Example images and audio signals from the multimodal *Materials* dataset

Data. The multimodal *Materials* dataset contains data from 26 textured plates made from 17 different material types. From each plate we recorded five audio signal with 44.1 kHz sampling frequency and segmented them into 450 overlapping sections of 50 ms, which we represented by phase and amplitude invariant *cepstral features* [19]. We clustered the resulting 58 500 feature vectors into an auditory codebook using k-means and represented each recording by a 1000-bin histogram, like in a bag-of-words representation. For the image data, we took high resolution photos with different in-plane rotations for a total of four to eight grayscale images per material. We computed *local binary patterns* over 8-neighborhoods considering only *uniform patterns* [21] such that any image region can be represented by a 58-dimensional histogram. Note that we intentionally chose a setup that is simple and easy to reproduce instead of a more powerful texture representation because our goal is not to improve the state of the art in texture classification but to examine the properties of multimodal feature extraction. To match the one-dimensional nature of the audio domain, we extracted single-pixel image strips with 16 pixel offset between them, resulting in a total of 32 histograms per image. For both, audio and visual data, we normalized each feature dimension to have zero mean and unit variance in order to reduce the influence of some histogram bins being more populated than others.

Experimental Setup. Our experiments reflect the situation where image and audio are present during dimensionality reduction itself, but only audio in the later application to new data. For this we split the data into two equally sized parts, called *context* and *task* data. We use WMCA to compute projection directions from the context data. As no perfect pairing between images and audio samples is available, we rely on the weak pairing information provided by the knowledge of which audio

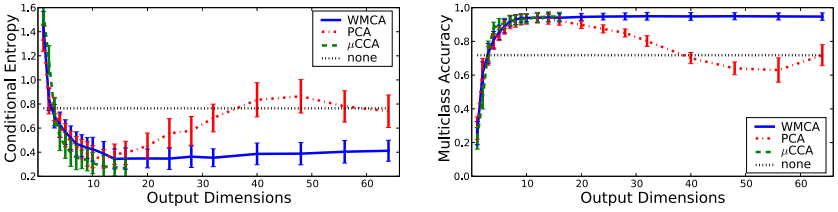


Fig. 2. Dimensionality reduction for unsupervised and supervised texture discrimination. The plots depict the conditional entropy (left, lower is better) and multi-class accuracy (right, higher is better) for different numbers of output dimensions.

signal was recording from which surface. In this linear bimodal case, each iteration of the WMCA algorithm takes only seconds. Convergence takes 2 to 50 iterations, depending on the output dimensionality.

We use the resulting dimensionality reduction functions to project the audio part of the task data to a new representation, and we measure the resulting clustering and classification performance. The unsupervised setup consists of applying *k-means* and measuring the quality of the resulting clusters by computing the *conditional entropy measure* [26,32] with respect to the ground truth. To simplify the setup we assume that the correct number of clusters is known a priori. In practical application, this number would have to be estimated from data. For the supervised setup, we measure the classification accuracy of a leave-one-out classifier; that is, for every point in the task set we determine its nearest neighbor and compute how often the labels of both samples coincide. For comparison we report the results of two baseline methods: unimodal dimensionality reduction with PCA that we apply separately to each modality, and fully-paired CCA, that is applicable when we use the data means of each weakly-paired group as input instead of the original samples (denoted μCCA). In addition we report the results without applying any dimensionality reduction.

Results. Figure 2 shows the results of the described procedure as mean and standard deviation over 100 random stratified splits of the data into context and task sets. We observe the same effect in both setups: all techniques identify the relevant output dimensions first and cause better results than when no dimensionality reduction is applied. However, when the number of output dimensions is increased, PCA starts to recover noise dimensions which decreases the performance, whereas WMCA's performance remains stable. Because μCCA uses the group means as inputs, it has only as many input samples as there are groups and therefore it cannot recover more than 17 output dimensions in this setup. In conclusion, the results of this section show that the main positive effect of using the multimodal dimensionality reduction in this case is improved noise suppression, which results in higher robustness in the choice of the number of output dimensions.

5.2 Transfer Learning

The previous experiments showed that WMCA is able to use multimodal data to infer which data dimensions are relevant and which are not. In this section we show how a similar effect can be used for *transfer learning* with attribute representations. Transfer learning consists of solving a learning task by making use of another, related, learning task, see [23] for a general overview and [22] for the specific case of transfer learning by dimensionality reduction. In our case, we want to improve the accuracy of an image classification system by making use of the data from another image classification task despite the fact that this has a disjoint set of classes and examples.

Data. For our experiments we use the *Animals with Attributes (AwA)*[2] dataset that has recently been introduced as a benchmark for attribute-based classification [15]. It consists of approximately 30,000 images of 50 animals classes as well as descriptions of the classes in terms of 85 binary semantic attributes, see Figure 3. The images are represented by the feature vectors that come with the dataset (based on SIFT, SURF, colorSIFT, local self similarity and color histogram features). We concatenate these into 10688-dimensional feature vectors and we remove the effect of inhomogeneous feature scaling by normalizing each dimension to zero mean and unit variance. The transformations necessary for this are saved in order to apply them to the task data later.

Experimental Setup. In our experiment largely follow the protocol of [15]. We split the set of classes into a context part consisting of forty classes and a task part consisting of ten classes. From the context data we chose 100 images per class, except for the *mole* category which has only 92 images that we use all, and we apply WMCA with the attribute representation as a second modality that is not available at test time. By assuming only a weak pairing between the domains, WMCA in particular is able to ignore outliers in the training set, whose actual image contents do not coincide well with the attribute vector. The quality of the resulting representation is determined by measuring the accuracy of a classifier for the task data. As baselines we again compute projection directions using PCA and CCA of the group means (μCCA). Because we assume that the context part has label information, we are able to also use LDA as a baseline. Additionally, we also include the case of not doing dimensionality reduction.

On the task set, we perform image classification in a Caltech-like setup. We randomly select a small number of training images per class, and classify a disjoint set of 30 randomly chosen test images using the nearest neighbor decision rule in the feature space induced by the projection directions found during the context stage. As in the case of texture discrimination our experimental setup is motivated by easy reproducibility. In particular we avoid free parameters that require model selection.

Results. Figure 4 shows the results for different numbers of training images and output dimensions as mean accuracy and standard error over 100 train/test

[2] Available for download at `http://attributes.kyb.tuebingen.mpg.de`

Fig. 3. Example images and attributes from the *Animals with Attributes* dataset

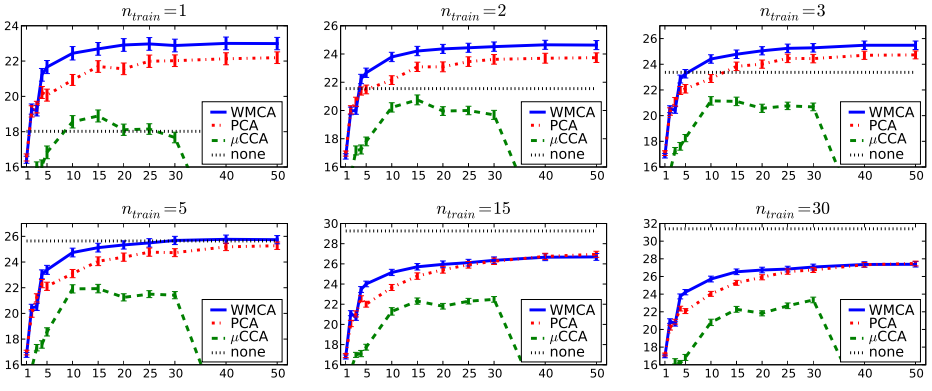


Fig. 4. Results of attribute-based transfer learning. The plots show the multi-class accuracy (y-axis) with n_{train} training images for different number of output dimensions (x-axis).

splits. When few training examples are available (top row), the representation found by WMCA leads to significantly higher classification accuracy than the representations obtained by PCA and also those by not using dimensionality reduction. When the number of training examples is increased WMCA is still superior to PCA when few output dimensions are wanted, but both are not able to exceed classification accuracy without dimensionality reduction anymore. This is consistent with the general observation that transfer learning works best in the regime when few training examples are available. However, dimensionality reduction can still be beneficial if runtime is an issue, as it makes the nearest neighbor lookup considerably faster than when the full features vectors are used.

μCCA leads to lower classification accuracy than both generative methods. Also, the performance does not improve any further when the number of output dimensions exceeds 10, which we interpret this as an overfitting effect. Because the data means provide only 40 data points, highly correlated directions can occur just due to noise effects. The plots in Figure 4 do not contain LDA, which never achieved classification accuracies that where significantly better than the chance level. The reason for this is LDA's discriminative objective. When applied to the context data it identifies projection directions that best encode the context class structure, but these do not reflect the class structure in the task set.

Overall, the results we achieve are comparable with previous work on the AwA dataset, which is known to be a difficult one. The most similar setup to ours is [29], where linear distance learning resulted in 23.7% accuracy in a one-shot setup, and a logistic representation in 27.2%. In [15], accuracies of 27.8% and 40.5% are reported, but based on a different test situation that made use of the attribute description at test time.

6 Conclusions

We have introduced weakly-paired maximum covariance analysis (WMCA) for multimodal dimensionality reduction. It overcomes the main limitation of MCA, from which it is derived, as it does not require fully paired data. Instead it treats missing pairings as latent variables which are inferred jointly with the projection directions. We showed how WMCA can be kernelized to perform non-linear dimensionality reduction. However, from a practical point of view, the most satisfactory setup is the linear two-modality case, where solving WMCA requires only two very efficient standard procedures: solving linear assignment problems and singular value decompositions.

In our experiments we illustrated two applications where multimodal dimensionality reduction was beneficial. In texture discrimination, WMCA produced more robust representations than the baselines. In transfer learning, when few training examples are available, WMCA was able to improve classification accuracy by transferring information from a context set to the main task.

Our initial experience with WMCA opens several directions for future work. Apart from practical application in robotics and video retrieval, we plan to derive more efficient techniques for applying kernelized WMCA at test time, e.g. based on reduced set methods and sparsification.

References

1. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: NIPS (2003)
2. Bach, F.R., Jordan, M.I.: A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Department of Statistics, University of California, Berkeley (2005)
3. Baudat, G., Anouar, F.: Generalized discriminant analysis using a kernel approach. Neural Computation 12(10) (2000)
4. Blaschko, M., Gretton, A.: Learning taxonomies by dependence maximization. In: NIPS (2009)
5. Blaschko, M., Lampert, C.H.: Correlational spectral clustering. In: CVPR (2008)
6. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of Eugenics 7 (1936)
7. Golub, G.H., Van Loan, C.F.: Matrix computations. Johns Hopkins Univ. Press, Baltinmore (1996)
8. Hardoon, D., Mourao-Miranda, J., Brammer, M., Shawe-Taylor, J.: Unsupervised analysis of fMRI data using kernel canonical correlation. NeuroImage 37(4) (2007)

9. Hardoon, D., Szedmak, S., Shawe-Taylor, J.: Canonical correlation analysis: an overview with application to learning methods. Neural Computation 16(12) (2004)
10. Hinton, G., Salakhutdinov, R.: Reducing the dimensionality of data with neural networks. Science 313(5786) (2006)
11. Hofmann, T.: Probabilistic latent semantic indexing. In: ACM SIGIR (1999)
12. Hotelling, H.: Relation between two sets of variates. Biometrika 28 (1936)
13. Jonker, R., Volgenant, A.: A shortest augmenting path algorithm for dense and sparse linear assignment problems. Computing 38(4) (1987)
14. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Research Logistics Quarterly 2 (1955)
15. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: CVPR (2009)
16. Lei, Z., Bai, Q., He, R., Li, S.Z.: Face shape recovery from a single image using CCA mapping between tensor spaces. In: CVPR (2008)
17. Lienhart, R., Romberg, S., Hörster, E.: Multilayer pLSA for multimodal image retrieval. In: CIVR (2009)
18. Livescu, K., Stoehr, M.: Multi-view learning of acoustic features for speaker recognition. In: Automatic Speech Recognition and Understanding (2009)
19. Logan, B.: Mel frequency cepstral coefficients for music modeling. In: International Symposium on Music Information Retrieval (2000)
20. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: 5th Berkeley Symposium on Mathematics Statistics and Probability (1967)
21. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. PAMI 24(7) (2002)
22. Pan, S.J., Kwok, J.T., Yang, Q.: Transfer learning via dimensionality reduction. In: AAAI (2008)
23. Pan, S.J., Yang, Q.: A survey on transfer learning. Knowledge and Data Engineering (2009)
24. Papadimitriou, C.H., Raghavan, P., Tamaki, H., Vempala, S.: Latent semantic indexing: A probabilistic analysis. Computer and System Sciences 61(2) (2000)
25. Pearson, K.: On lines and planes of closest fit to systems of points in space. Philosophical Magazine Series 6. 2(11) (1901)
26. Rosenberg, A., Hirschberg, J.: V-measure: A conditional entropy-based external cluster evaluation measure. In: EMNLP-CoNLL (2007)
27. Rosipal, R., Trejo, L.J.: Kernel partial least squares regression in reproducing kernel Hilbert space. JMLR 2 (2002)
28. Schölkopf, B., Smola, A.J.: Learning with kernels. MIT Press, Cambridge (2002)
29. Tang, K., Tappen, M., Sukthankar, R., Lampert, C.H.: Optimizing one-shot recognition with micro-set learning. In: CVPR (2010)
30. Tenenbaum, J.B., Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290(5500) (2000)
31. Tucker, L.R.: An inter-battery method of factor analysis. Psychometrika 23 (1958)
32. Tuytelaars, T., Lampert, C.H., Blaschko, M.B., Buntine, W.: Unsupervised object discovery: A comparison. IJCV 88(2) (2010)
33. Wold, H.: Estimation of principal components and related models by iterative least squares. Multivariate Analysis 1 (1966)

Optimizing Complex Loss Functions in Structured Prediction

Mani Ranjbar, Greg Mori, and Yang Wang

School of Computing Science
Simon Fraser University, Canada

Abstract. In this paper we develop an algorithm for structured prediction that optimizes against *complex* performance measures, those which are a function of false positive and false negative counts. The approach can be directly applied to performance measures such as F_β score (natural language processing), intersection over union (image segmentation), Precision/Recall at k (search engines) and ROC area (binary classifiers). We attack this optimization problem by approximating the loss function with a piecewise linear function and relaxing the obtained QP problem to a LP which we solve with an off-the-shelf LP solver. We present experiments on object class-specific segmentation and show significant improvement over baseline approaches that either use simple loss functions or simple compatibility functions on VOC 2009.

1 Introduction

Solving challenging vision problems such as image understanding, image segmentation, and video retrieval arguably requires the use of "complex" structured models – those incorporating relationships between multiple input and output entities. Evidence for this comes from state-of-the-art approaches to the aforementioned problems. For example, Hoiem et al. [1] formulate image understanding models that tie together object locations, camera parameters, and surfaces. Blaschko and Lampert [2] localize objects using an efficient solution to a structured output regression model. Desai et al. [3] learn models for simultaneously detecting all objects in an image. Non-max suppression and contextual object co-occurrence statistics are learned in a discriminative fashion. Image segmentation is a canonical example of structured labeling problem (e.g. [4,5,6]).

For many of these problems the natural performance measures are also "complex" – ones that do not decompose into a simple sum of individual terms measured over each output entity. Examples of such measures are object detection scores that penalize for multiple detections on a single true positive (e.g PASCAL VOC [7]) and region labeling or object segmentation scores that penalize for over and under labeling or segmentation (e.g. intersection / union score). Typical methods for solving these problems learn parameters against other performance measures, e.g. Hamming loss for segmentation, and then apply post-processing techniques (e.g. non-maximum suppression in object detection) to address the

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 580–593, 2010.

structure in the performance measure. Instead, in this paper we develop an algorithm for linking these two together and formulate learning as jointly considering the complex, structured relationships between output variables in the model and in the learning objective.

The main contribution of this paper is developing a general algorithm for addressing this type of learning problem with complex models and those complex loss functions which are a function of false positive and false negative counts. We specifically apply it to image segmentation, but note that the algorithm can be applied more broadly. We experiment with a standard Markov Random Field (MRF) segmentation model that contains both unary terms for labeling pixels and pairwise terms on the labels of neighbouring pixels. We show that learning the parameters to this model under an objective directly tied to the performance measure significantly improves performance relative to baseline algorithms on the PASCAL VOC Segmentation Challenge.

2 Previous Work

A wide range of learning algorithms exist. Despite technical differences, all of these approaches rely on a performance measure to define what is a "good" result. Based on the complexity of the performance measure, two general approaches to optimize it are imaginable, formulate the learning problem to directly optimize this measure, or approximate this measure with a simpler one and try to optimize it aiming to indirectly optimize the original complex performance measure. We will call the former "direct optimization" and the latter "indirect optimization".

Due to the complexity of some performance measures, e.g., average precision and intersection over union, many state-of-the-art approaches in different challenges exploit an indirect optimization. Looking at PASCAL VOC challenge 2009 [7], for example, average precision and intersection over union are defined as performance measures for detection and segmentation tasks respectively, but methods for both tasks use indirect optimizations for solving these problems.

Structured prediction has become popular in computer vision. Taskar et al. [8] and Tsochantaridis et al. [9] have the same formulation for structured prediction using a max-margin criterion. Both of them need to solve the "most violated constraint" [9], or loss augmented inference [8] in each iteration of their gradient descent to find the optimal parameters. They assume the loss function is decomposable and therefore solving for the most violated constraint is as hard as doing the inference without the loss function, which is assumed to be tractable. Joachims [10] proposed an approach to efficiently compute the most violated constraint when the loss function is not decomposable, but limited the underlying model by allowing only simple compatibility functions, those which involve only a single input and output. In this paper we provide an algorithm for structured prediction with a complex compatibility function that optimizes against complex performance measures, those which are a function of false positive and false negative counts.

3 Background

To create a foundation for the proposed approach, we start with an overview of our learning formulation. Next, we discuss the two common approaches, one based on a simple loss function with a complex compatibility function and the other with complex loss function and simple compatibility function. We call a loss function simple if it can be decomposed into loss on individual training samples. Likewise, a compatibility function is called simple if it only depends on a single sample point and its ground-truth label. Finally, we propose a framework to incorporate certain complex loss functions and complex compatibility functions in structured prediction.

3.1 Problem Formulation

The goal of our learning problem is defined as finding a function $h \in \mathcal{H}$ from the hypothesis space $\mathcal{H}$ given training samples $\mathrm{S} = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N))$ that optimizes the expected prediction performance on the new samples S' of size n'.

$$R^{\Delta}(h) = \int \Delta((h(\boldsymbol{x}'_1), h(\boldsymbol{x}'_2), \ldots, h(\boldsymbol{x}'_{n'})), (y'_1, y'_2, \ldots, y'_{n'}))dPr(S'). \tag{1}$$

In general, the loss function Δ cannot be decomposed into a linear combination of a loss function δ over individual samples. But, for simplicity, most discriminative learning algorithms (e.g. SVM) assume decomposibility and i.i.d. samples, which allows for rewriting Eq. 1 as

$$R^{\Delta}(h) = R^{\delta}(h) = \int \delta(h(\boldsymbol{x}'), y')dPr(\boldsymbol{x}', y'). \tag{2}$$

Instead of solving the estimated risk in Eq. 2, learning algorithms approximate that with empirical risk $\hat{R}^{\delta}$ defined as

$$\hat{R}^{\delta}(h) = \frac{1}{n}\sum_{i=1}^{N} \delta(h(\boldsymbol{x}_i), y_i). \tag{3}$$

For non-decomposable loss functions, such as F_1 score or intersection over union, optimizing Eq. 2 does not provide the desired answer. Rather, we are interested in finding an algorithm that can directly optimize the empirical risk based on the sample loss,

$$\hat{R}^{\Delta}_{S}(h) = \Delta((h(\boldsymbol{x}_1), h(\boldsymbol{x}_2), \ldots, h(\boldsymbol{x}_n)), (y_1, y_2, \ldots, y_n)). \tag{4}$$

Note that finding an $h \in \mathcal{H}$ that optimizes Eq. 4 for an arbitrary loss function Δ can be computationally challenging.

3.2 Structured Prediction Learning

For non-decomposable loss functions, one can reformulate the SVM based on the idea of multivariate prediction [10]. Instead of having a mapping function

$h : \mathcal{X} \rightarrow \mathcal{Y}$ from a single example $\boldsymbol{x}$ to its label y, where $\boldsymbol{x} \in \mathcal{X}$ and $y \in \{-1, +1\}$, we look at all examples at once and try to learn a mapping function $\bar{h} : \mathcal{X} \times \cdots \times \mathcal{X} \rightarrow \bar{\mathcal{Y}}$, where $\bar{\mathcal{Y}} \in \{-1, +1\}^N$. We define $\bar{\boldsymbol{x}} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$, and $\boldsymbol{y} = (y_1, \ldots, y_N)$.

We can define the best labeling using a linear discriminant function

$$\bar{h}(\bar{\boldsymbol{x}}) = \arg\max_{\boldsymbol{y}' \in \bar{\mathcal{Y}}} \boldsymbol{w}^T \Psi(\bar{\boldsymbol{x}}, \boldsymbol{y}'). \tag{5}$$

Here, function Ψ measures the compatibility of the data points and their assigned labels. If we define the Ψ function as a simple form

$$\Psi(\bar{\boldsymbol{x}}, \boldsymbol{y}') = \sum_{i=1}^{N} y_i' \boldsymbol{x}_i, \tag{6}$$

that only depends on individual training points and their labels, the optimal labeling sequence is

$$\arg\max_{\boldsymbol{y}' \in \bar{\mathcal{Y}}} \boldsymbol{w}^T \Psi(\bar{\boldsymbol{x}}, \boldsymbol{y}') = \arg\max_{\boldsymbol{y}' \in \bar{\mathcal{Y}}} \sum_{i=1}^{N} y_i' \boldsymbol{w}^T \boldsymbol{x}_i = (h(\boldsymbol{x}_1), \ldots, h(\boldsymbol{x}_N)), \tag{7}$$

which is exactly the same as the optimal labeling in SVM.

One way of incorporating a loss function Δ in SVM formulation is *Margin Rescaling* [9],

$$\min_{\boldsymbol{w}, \xi \geq 0} \|\boldsymbol{w}\|^2 + C\xi \tag{8}$$

$$s.t.\ \forall \boldsymbol{y}' \in \bar{\mathcal{Y}} \backslash \boldsymbol{y},\ \boldsymbol{w}^T \left[\Psi(\bar{\boldsymbol{x}}, \boldsymbol{y}) - \Psi(\bar{\boldsymbol{x}}, \bar{y}')\right] \geq \Delta(\boldsymbol{y}, \boldsymbol{y}') - \xi \tag{9}$$

Similar to the original SVM formulation, ξ in Eq. 8 is an upper bound on $\Delta(\bar{h}(\bar{\boldsymbol{x}}), \boldsymbol{y})$ [10].

The guarantee for convergence in polynomial time, the potential for incorporating complex loss functions in the objective and good performance in practice are the most important reasons why structured prediction has garnered much attention in computer vision recently.

In the standard approaches for solving Eq. 8, the output vector, $\tilde{\boldsymbol{y}}$, corresponding to the most violated constraint should be found repeatedly [9],

$$\tilde{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}' \in \bar{\mathcal{Y}}} \Delta(\boldsymbol{y}, \boldsymbol{y}') + \boldsymbol{w}^T \Psi(\bar{\boldsymbol{x}}, \boldsymbol{y}'). \tag{10}$$

Finding $\tilde{\boldsymbol{y}}$ is computationally challenging given an arbitrary loss function, $\Delta(\boldsymbol{y}, \boldsymbol{y}')$, and compatibility function, $\Psi(\bar{\boldsymbol{x}}, \boldsymbol{y}')$. However, solving Eq. 10 in two special cases has been shown to be efficient. We categorize these approaches based on the simplicity of their Δ and Ψ functions. We call a loss function simple if it can be decomposed into individual training samples. Likewise, a compatibility function is called simple if it decomposes over single sample points and their ground-truth labels.

3.3 Simple Δ, Complex Ψ

Optimizing the parameters of a MRF structure when the loss function can be decomposed into the loss of individual samples falls into this category. One popular application in this category is foreground-background segmentation with Hamming loss, which is defined as

$$\Delta_H = \sum_i \mathbb{1}_{[y_i \neq y'_i]}. \tag{11}$$

Szummer et al. [6] have employed this formulation and reported promising results for interactive segmentation.

Decomposibility of the loss function results in a MRF form for Eq. 10, because the loss function can be treated as another unary term that adds up to the unary terms of the compatibility function. Assuming binary labels, this MRF can be solved efficiently using graphcut.

The advantage of this approach is to exploit pairwise connections, but it is only tractable for decomposable loss functions.

3.4 Complex Δ, Simple Ψ

The other special case presented by Joachims [10], is when the Ψ function has a simple form of

$$\Psi(\bar{\boldsymbol{x}}, \boldsymbol{y}') = \sum_{i=1}^{N} y'_i \boldsymbol{x}_i. \tag{12}$$

If the loss function, Δ, is just a function of true positive (TP), false positive (FP) and false negative (FN), then there are at most $N_p \times N_n$ distinct loss values, where N_p and N_n represent the number of positive and negative training examples, respectively. Hence, Eq. 10 can be solved by iterating over all loss values and maximizing $\boldsymbol{w}^T \Psi(\boldsymbol{x}, \boldsymbol{y}')$ subject to the value of TP, FP and FN [10].

Unlike the approach of Szummer et al. [6], many standard accuracy measures that lead to non-decomposable loss functions, such as F_β score (natural language processing), intersection over union (image segmentation), Precision/Recall at k (web search engines) and ROC area (binary classifiers) can be directly optimized by this approach. However, this method cannot benefit from the pairwise interactions of training samples, which are shown to be advantageous in many applications, such as object detection [3] and scene interpretation [1].

4 Proposed Approach: Solving Complex Δ, Complex Ψ

Discussing the advantages and shortcomings of the previous methods, we now propose an approach to directly optimize certain complex loss functions in a Markov network. Here, we can optimize non-decomposable accuracy measures, such as F_β and intersection over union and still be able to benefit from pairwise interactions between training points.

We choose to follow the general framework of Structural$_{\text{SVM}}$ [9], shown in Eq. 8. Solving Eq. 8 requires finding the most violated constraint (Eq. 10) at each iteration and modifying the parameter vector $\boldsymbol{w}$ accordingly [9]. We propose a novel method to efficiently solve for an approximate most violated constraint for certain non-decomposable loss functions in presence of pairwise terms in the compatibility function, Ψ.

We can summarize the proposed approach as

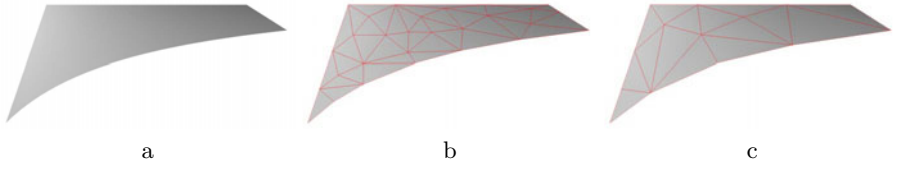
1. Replacing the original non-decomposable loss function with a piecewise linear approximation,
2. Writing the problem of finding the most violated constraint as a quadratic program,
3. Converting the quadratic program to a linear program and solve the relaxed problem.

4.1 Piecewise Linear Approximation

Many standard accuracy measures, including the one presented in the previous section, share the property that they can be computed from the contingency table[1]. Given the number of positive and negative examples, N_p and N_n, the loss function corresponding to these accuracy measures is just a function of FP and FN. Using piecewise linear approximation, we can write

$$\Delta(FP, FN) \simeq \tilde{\Delta}(FP, FN) = \sum_{j=1}^{M} \mathbb{1}_{[(FP,FN)\in\mathfrak{R}_j]} \{\alpha_j FP + \beta_j FN + \gamma_j\} \quad (13)$$

where, M is the number of subregions (pieces), α_j, β_j and γ_j represent the j^{th} plane coefficients and $\mathfrak{R}_j$s are the subregions that partition the space spanned by FP and FN.



a b c

Fig. 1. Intersection over union loss surface in FP and FN space. a) Exact surface, b) a piecewise linear approximation with 40 subregions, c) a piecewise linear approximation with 15 subregions.

As an example, Figure 1 illustrates the intersection over union loss function,

$$\Delta_{\frac{\cap}{\cup}}(FP, FN) = \frac{FN + FP}{N_p + FP}, \quad (14)$$

along with its piecewise linear approximations using 15 and 40 pieces.

Given the subregion $\mathfrak{R}_j$, the original non-linear loss function is a linear function of FP and FN. The next step is to substitute the approximated loss function, $\tilde{\Delta}$ into Eq. 10 and solve for the most violated constraint.

[1] Is just a function of TP, FP, TN and FN.

4.2 Forming the Quadratic Program

Choosing the right form of Ψ function is crucial to achieve high performance. In segmentation, for example, employing only unary terms in the Ψ function that model the relationship between an observed pixel and its label result in a lack of smoothness in the labeling. Hence, methods usually incorporate pairwise terms in the Ψ function to smooth the output labeling. We define our Ψ with unary and pairwise terms as

$$\Psi(\bar{\boldsymbol{x}}, \boldsymbol{y}) = \sum_i (2y_i - 1)\phi_u(\boldsymbol{x}_i) + \sum_i \sum_{j \in \mathcal{N}_i} (y_i + y_j - 2y_i y_j)\phi_p(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{15}$$

Here $\mathcal{N}_i$ is the set of neighbors of sample i and we have assumed $y \in \{0, 1\}$. We rewrite Eq. 10 with approximated loss function, $\tilde{\Delta}$ as

$$\tilde{\boldsymbol{y}}* = \arg\max_{\boldsymbol{y}' \in \bar{\mathcal{Y}}} \tilde{\Delta}(\boldsymbol{y}, \boldsymbol{y}') + \boldsymbol{w}^T \Psi(\bar{\boldsymbol{x}}, \boldsymbol{y}') \tag{16}$$

$$= \arg\max_{\boldsymbol{y}' \in \bar{\mathcal{Y}}} \tilde{\Delta}(\boldsymbol{y}, \boldsymbol{y}') + \boldsymbol{w}_u^T \sum_i (2y_i' - 1)\phi_u(\boldsymbol{x}_i) \tag{17}$$

$$+ \boldsymbol{w}_p^T \sum_i \sum_{j \in \mathcal{N}_i} (y_i' + y_j' - 2y_i' y_j')\phi_p(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

where $\boldsymbol{w} = [\boldsymbol{w}_u; \boldsymbol{w}_p]$ (concatenation of the two).

Note that $FP = \sum_i (1 - y_i)y_i'$ and $FN = \sum_i y_i(1 - y_i')$, where y_i is the true label and y_i' is the predicted label for the i^{th} example. If we assume that the loss values fall in subregion $\mathfrak{R}_k$, we can write Eq. 17 as

$$\tilde{\boldsymbol{y}}* = \arg\max_{\boldsymbol{y}' \in \bar{\mathcal{Y}}} \Bigg(\alpha_k \sum_i (1 - y_i)y_i' + \beta_k \sum_i y_i(1 - y_i') + \gamma_k + \tag{18}$$

$$\boldsymbol{w}_u^T \sum_i (2y_i' - 1)\phi_u(\boldsymbol{x}_i) + \boldsymbol{w}_p^T \sum_i \sum_{j \in \mathcal{N}_i} (y_i' + y_j' - 2y_i' y_j')\phi_p(\boldsymbol{x}_i, \boldsymbol{x}_j) \Bigg).$$

Note that Eq. 18 only includes the predicted label y' in linear and quadratic forms. Hence, we can write a quadratic program based on Eq. 18 subject to the loss values being in subregion $\mathfrak{R}_k$,

Maximize:

$$\alpha_k \sum_i (1 - y_i)y_i' + \beta_k \sum_i y_i(1 - y_i') + \gamma_k + \tag{19}$$

$$\sum_i (2y_i' - 1)[\boldsymbol{w}_u^T \phi_u(\boldsymbol{x}_i)] + \sum_i \sum_{j \in \mathcal{N}_i} (y_i' + y_j' - 2y_i' y_j')[\boldsymbol{w}_p^T \phi_p(\boldsymbol{x}_i, \boldsymbol{x}_j)]$$

Subject to:

$$\left\{ \sum_i (1 - y_i)y_i', \sum_i y_i(1 - y_i') \right\} \in \mathfrak{R}_k, \quad i = 1, \ldots, N \tag{20}$$

In order to have linear constraints in Eq. 20, the boundary of all subregions should be definable as a linear function of $\boldsymbol{y}'$. One way is to separate the subregions by straight lines. If for example, we partition the space spanned by FP and FN into triangles (Fig. 1b,c) then Eq. 20 will be substituted by three linear constraints corresponding to the three sides of the triangle.

4.3 Converting Quadratic Program to Linear Program

The quadratic program in Eq. 19 is potentially non-convex, since there is no constraint on the coefficients of the objective function. So, instead of looking for a local optima of this non-convex function, we relax the problem (MAP-MRF LP relaxation [11]) by introducing some variables that substitute the quadratic terms in the objective function and form a linear program, which is convex. In detail, we introduce four new variables corresponding to four different possible configurations of a pair of labels as follows.

$$\mathfrak{y}_{ij}^{00} \equiv (1-y_i')(1-y_j'),\ \mathfrak{y}_{ij}^{01} \equiv (1-y_i')y_j',\ \mathfrak{y}_{ij}^{10} \equiv y_i'(1-y_j'),\ \mathfrak{y}_{ij}^{11} \equiv y_i'y_j'. \tag{21}$$

We also add a set of constraints to relate the introduced variables to y' variables. The final linear program is

Maximize:

$$\alpha_k \sum_i (1-y_i)y_i' + \beta_k \sum_i y_i(1-y_i') + \gamma_k + \tag{22}$$
$$\sum_i (2y_i'-1)[\boldsymbol{w}_u^T \phi_u(\boldsymbol{x}_i)] + \sum_i \sum_{j \in \mathcal{N}_i} (\mathfrak{y}_{ij}^{01} + \mathfrak{y}_{ij}^{10})[\boldsymbol{w}_p^T \phi_p(\boldsymbol{x}_i, \boldsymbol{x}_j)]$$

Subject to:

$$\left\{ \sum_i (1-y_i)y_i', \sum_i y_i(1-y_i') \right\} \in \mathfrak{R}_k, \quad i = 1, \ldots, N, j \in \mathcal{N}_i \tag{23}$$
$$\mathfrak{y}_{ij}^{10} + \mathfrak{y}_{ij}^{11} = y_i' \tag{24}$$
$$\mathfrak{y}_{ij}^{01} + \mathfrak{y}_{ij}^{11} = y_j' \tag{25}$$
$$\mathfrak{y}_{ij}^{00} + \mathfrak{y}_{ij}^{01} + \mathfrak{y}_{ij}^{10} + \mathfrak{y}_{ij}^{11} = 1 \tag{26}$$

Solving this LP for thousands of binary variables (labels), is not computationally tractable. So instead we relax the label values to real numbers between zero and one and solve for optimal labeling. Later, we map the optimal labels to binary values by rounding the results. We solve Eq. 22 for each subregion separately, and return the labeling of the one with the maximum objective as the most violated constraint.

5 Experiments

As a concrete example, we experiment on object segmentation using our proposed approach. Given an input image, the goal is to produce a 0/1 mask, in

which a pixel gets label 1 if it is part of a given object category and label 0 otherwise.

Dataset. We run our experiments on the VOC2009 Segmentation [7] dataset. There are 749 images in the training set, 750 images in the validation set and 750 images in the test set. We train the parameters on the training set and evaluate performance on the validation set so that we can directly compare to baseline methods without relying on the VOC server. We compare the results using the intersection over union accuracy measure on 6 out of 20 object categories that can be localized the best employing our top-down features. Note that we perform the experiments on these objects independently. For example, when we segment object class car, any other object is taken as background. This is different from the VOC segmentation challenge in which the segmentation result should contain all object classes simultaneously. To combine our independent segmentations, we would need to have a score for each foreground pixel. Then, we could assign a pixel the label with maximum score. One way of scoring labels is the approach of Kohli [12] that can exactly compute the min marginals for graph cuts, however it is outside the scope of this paper.

Features. We define an MRF segmentation model with unary and pairwise features, for which the approximate inference is performed using FastPD [13]. Instead of working on the pixel level we first group the pixels into superpixels, which are fewer and therefore makes the learning process faster. Also they are larger so can be represented by more meaningful features. We use the superpixel extractor of Felzenszwalb et al. [14] that has three parameters. We set these parameters as $k = 200$, $MinArea = 1330$ and $\sigma = 0.01$. This setting of parameters result in an average of 50 superpixels per image of size 300×500 pixels.

To represent each superpixel, we use a set of bottom-up and top-down features, which form $\phi_u(\boldsymbol{x}_i)$ for superpixel i in Eq. 22. To create the bottom-up features, we compute Color SIFT features [15] on a dense grid with 6 pixel spacing in horizontal and vertical directions. We then turn this into a bag-of-words representation using a codebook of 1000 visual words.

For top-down features, we take a similar approach to the implicit shape model [16]. We first learn two appearance models for each of the 6 object categories using the detector of Felzenszwalb et al. [17]. The result includes two root filters and 6×2 part filters, where each root filter and 6 corresponding part filters model the object appearance in one pose. We run this detector on the training set and collect all bounding boxes that have positive scores. We then crop the ground-truth images on the bounding box locations and compute the average shape for the roots and parts, Fig 2.

We explain the rest of the process for one part, but the same process is applied to all parts and both roots. We find the potential part locations and their confidences by running the detector on the image in different scales. We call the result at each scale a confidence map, Fig. 3-b. Each potential part location casts its vote for the shape of that part proportional to its confidence. We implement this by convolving the confidence maps (different scales) with the

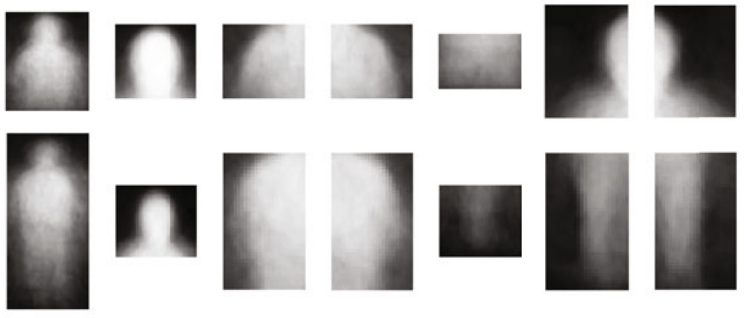


Fig. 2. Visualization of the average root and part shapes for person category. Each row corresponds to shape models obtained from root and part appearance models of one object pose.

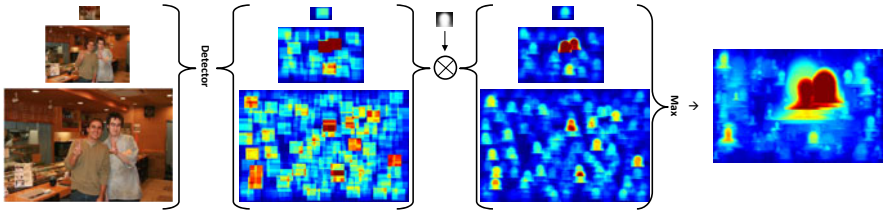


Fig. 3. The process of computing top-down features. Instead of showing the center of the detected parts we depict the bounding box for visualization purposes in the second stage.

average shape for that particular part. We call the convolution result in each scale a potential mask, Fig. 3-c. To merge the potential masks, we rescale them to the original image size and get the maximum of the masks, Fig. 3-d. We accumulate the mask values inside each superpixel to form the top-down feature corresponding to the part. Fig. 3 depicts the entire process for one part.

To employ the pairwise interaction between neighboring superpixels i and j, we define a set of pairwise features that represent $\phi_p(\boldsymbol{x}_i, \boldsymbol{x}_j)$ in Eq. 22. We first convert the image from RGB to La^*b^* color space. We define L_i, a_i and b_i to be the average L, a and b values inside superpixel i, respectively and assign the length of the common boundary between superpixel i and j to $\mathcal{P}_{ij}$. We then compute the pairwise features as

$$\boldsymbol{\phi}_p(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathcal{P}_{ij}. \exp\left[-\tau_1(L_i - L_j)^2, -\tau_2(a_i - a_j)^2, -\tau_2(b_i - b_j)^2\right]. \quad (27)$$

In our experiments the values of τ_1 and τ_2 are set to 2×10^{-2} and 5×10^{-3}, respectively.

Results. We compare the proposed method to two other methods based on their intersection over union segmentation accuracy. We use the same set of features for all methods. All three methods share the same general framework as explained by Tsochantaridis et al. [9]. The difference is in the form of their loss function Δ and their compatibility function Ψ. The first approach $BL1$ uses a decomposable Hamming loss function and a complex Ψ function including pairwise terms. The second method $BL2$ has been presented by Joachims [10] that can optimize a non-decomposable loss function, intersection over union in our experiment, but only includes unary terms in the Ψ function. And finally, the third method is the proposed approach that approximates the intersection over union loss function and can handle Ψ functions with unary and pairwise terms. We also show some segmentation results in Fig. 5 for all 6 object categories.

We use the same regularizer coefficient $C = 1$ for all three methods and set the number of subregions, M, for our piecewise linear approximation to 40. First, we triangulate the loss surface in FP, FN space finely. Then, we simplify the mesh into 40 triangles using a software called "Polygon Cruncher", which tries to approximate the original mesh as close as possible. To solve the LP problem of Eq. 22, we employ an off-the-shelf LP solver, Mosek [18].

Table 1. Intersection over union accuracies for 6 object categories

	BL1	BL2	Proposed Method
	Δ = Adjusted Hamming	$\Delta = \frac{\cap}{\cup}$	$\Delta = \frac{\cap}{\cup}$
	Unary + Pairwise	Unary	Unary + Pairwise
person	20.73	26.7	32.53
bus	25.49	22.65	31.69
aeroplane	21.23	12.65	32.11
car	23.37	22.86	27.83
horse	0.0	5.2	13.85
tv/monitor	2.24	6.63	12.69

In the training set, the number of superpixels that belong to the object are far fewer than the number of background superpixels, e.g., 1 foreground superpixel for every 25 background superpixels in person category. It means that reporting all superpixels as background gives Hamming score of $\frac{24}{25}$ or 96%. However, the same result obtains zero score based on intersection over union, because the intersection is simply empty. Therefore, we use *adjusted Hamming loss* defined as

$$\Delta_{AH} = \kappa FP + FN. \tag{28}$$

By changing κ we can adjust the relative contribution of foreground and background labels. In our experiment we set κ for each object to the ratio of

foreground and background superpixels in the training set. Without this adjustment $BL1$ would always return every superpixel as background.

The results reported in Table 1 show significant improvement in segmentation accuracy by the proposed method. Moreover, the results of $BL1$ and $BL2$ are comparable in a sense that in half of the categories $BL1$ performs better than $BL2$ and performs worse in the other half.

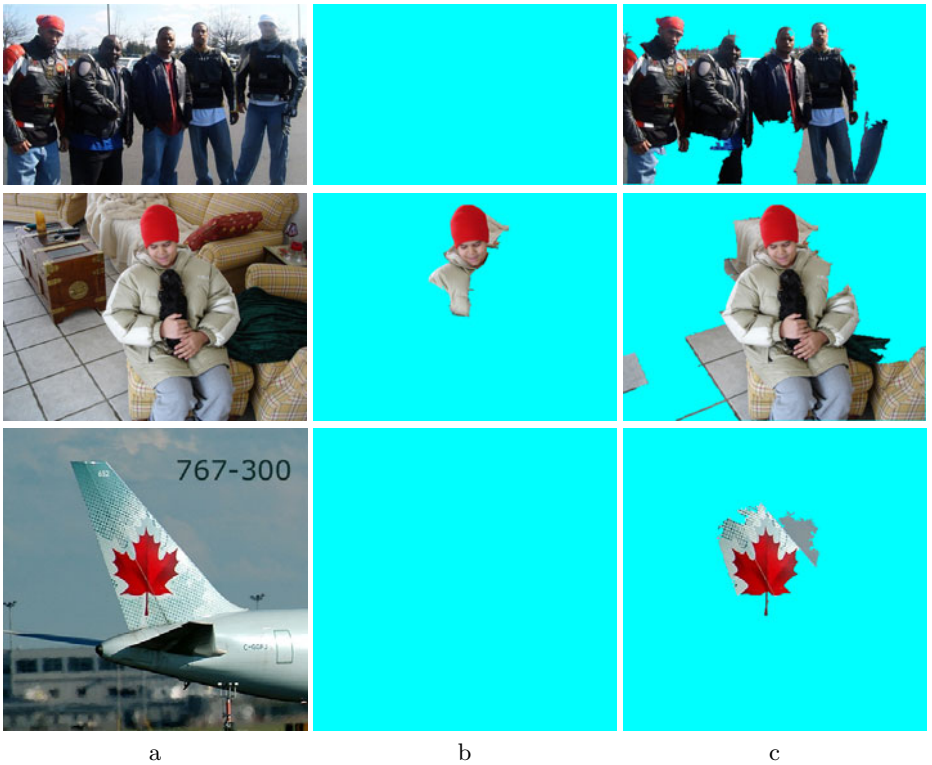


Fig. 4. Segmentation for person category. Optimizing adjusted Hamming loss (BL1) against our proposed method. a) input image, b) segmentation considering adjusted Hamming loss (BL1), c) our proposed method employing intersection over union. Intersection over union provides more true positives by possibly creating some false positives. Adjusted Hamming loss decreases false positive by sacrificing some true positives.

We compare the effect of optimizing adjusted Hamming loss versus intersection over union in Fig. 5. Adjusted Hamming loss tends to return fewer false positives, but with the cost of missing many true positives. In fact, it often marks all pixels as background, while intersection over union actually produces segmentations.

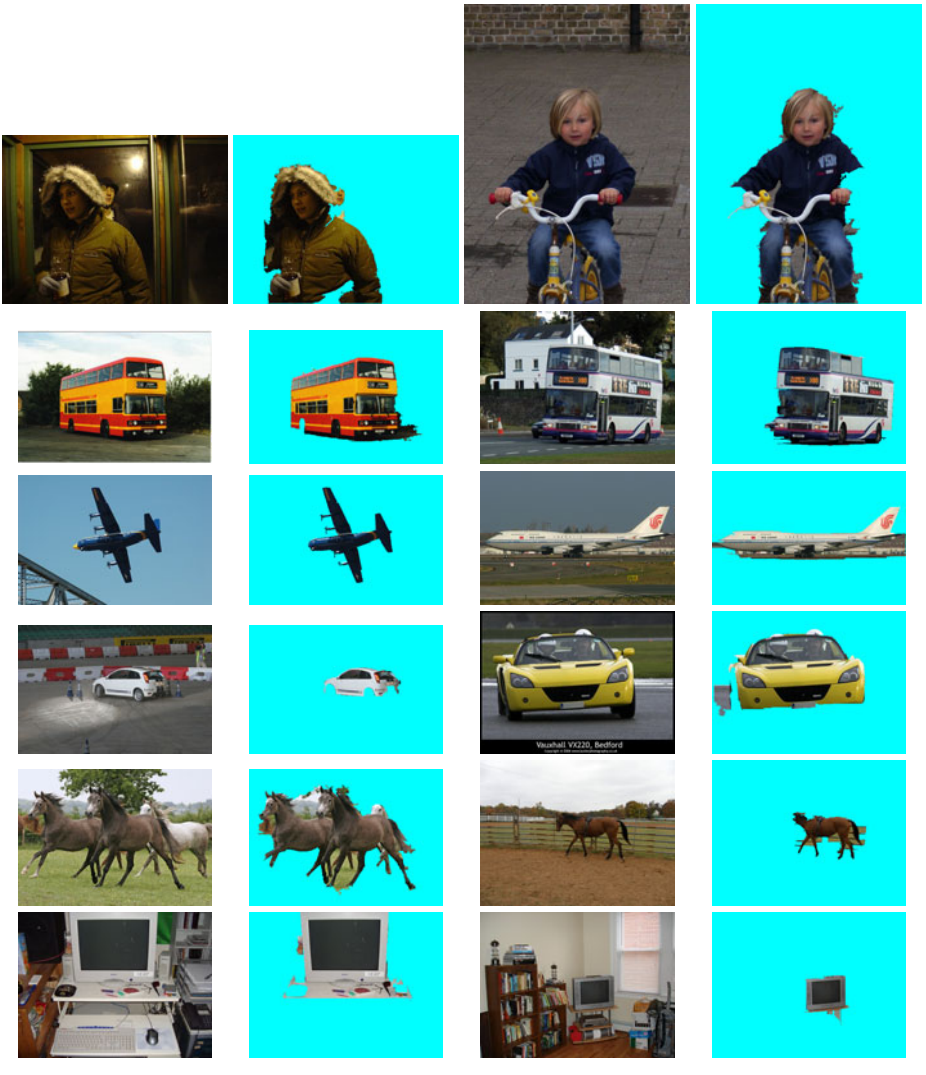


Fig. 5. Some segmentation results. Each row corresponds to one object category.

6 Conclusion

In this paper we develop a general algorithm for addressing learning problems with complex models and complex loss functions, those which are a function of false positive and false negative counts. We replace the original non-decomposable loss function with a piecewise linear approximation, and solve it using a linear programming relaxation of the original quadratic program. In future work it would be interesting to analyze the quality of these approximations. However, in this work we have provided experimental evidence of their effectiveness. In particular

we apply this method to learning an image segmentation model that contains both unary terms for labeling pixels and pairwise terms on the labels of neighbouring pixels. We show that learning the parameters to this model under an objective directly tied to the performance measure significantly improves performance relative to baseline algorithms on the PASCAL VOC Segmentation Challenge.

References

1. Hoiem, D., Efros, A.A., Hebert, M.: Closing the loop in scene interpretation. In: Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn. (2008)
2. Blaschko, M.B., Lampert, C.H.: Learning to localize objects with structured output regression. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 2–15. Springer, Heidelberg (2008)
3. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for multi-class object layout. In: ICCV (2009)
4. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and texture analysis for image segmentation. Int. Journal of Computer Vision 43, 7–27 (2001)
5. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. PAMI 23, 1222–1239 (2001)
6. 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)
7. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge (VOC2009) Results (2009), http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html
8. Taskar, B., Chatalbashev, V., Koller, D., Guestrin, C.: Learning structured prediction models: a large margin approach. In: ICML 2005, pp. 896–903 (2005)
9. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: ICML (2004)
10. Joachims, T.: A support vector method for multivariate performance measures. In: ICML 2005, pp. 377–384. ACM, New York (2005)
11. Werner, T.: A linear programming approach to max-sum problem: A review. IEEE Trans. PAMI 29, 1165–1179 (2007)
12. Kohli, P., Torr, P.H.S.: Measuring uncertainty in graph cut solutions. Comput. Vis. Image Underst. 112, 30–38 (2008)
13. Komodakis, N., Tziritas, G.: Approximate labeling via graph-cuts based on linear programming. IEEE Trans. PAMI 29 (2007)
14. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. Int. Journal of Computer Vision 59 (2004)
15. van de Sande, K.E.A., Gevers, T., Snoek, C.G.M.: Evaluating color descriptors for object and scene recognition. IEEE Trans. PAMI (2010)
16. 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, pp. 17–32 (2004)
17. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE Trans. PAMI (2009)
18. Mosek: The mosek optimization software (2010), http://www.mosek.com

A Novel Parameter Estimation Algorithm for the Multivariate t-Distribution and Its Application to Computer Vision

Chad Aeschliman, Johnny Park, and Avinash C. Kak

Purdue University
http://rvl.ecn.purdue.edu

Abstract. We present a novel algorithm for approximating the parameters of a multivariate t-distribution. At the expense of a slightly decreased accuracy in the estimates, the proposed algorithm is significantly faster and easier to implement compared to the maximum likelihood estimates computed using the expectation-maximization algorithm. The formulation of the proposed algorithm also provides theoretical guidance for solving problems that are intractable with the maximum likelihood equations. In particular, we show how the proposed algorithm can be modified to give an incremental solution for fast online parameter estimation. Finally, we validate the effectiveness of the proposed algorithm by using the approximated t-distribution as a drop in replacement for the conventional Gaussian distribution in two computer vision applications: object recognition and tracking. In both cases the t-distribution gives better performance with no increase in computation.

1 Introduction

Probability models are used in a wide range of applications in order to account for the uncertainty of processes and observations in a principled way. Often the true distribution underlying a process or observation is unknown or is difficult to use. In these cases one option is to use a nonparametric distribution. However, nonparametric distributions require a large amount of data to train, particularly in high-dimensional spaces. A common alternative is to fit a generic parametric probability model to the data.

By far the most commonly used parametric probability model is the multivariate Gaussian distribution. The Gaussian distribution is easy to use and has a number of nice properties. Parameter estimation for the Gaussian distribution is straightforward since its sufficient statistics are the parameters. Also, it is very easy to compute the marginal and conditional distributions from the joint distribution. However, for many applications the Gaussian distribution has tails which are too light; it tends to underestimate the probability of rare events occurring, which is unrealistic and can have a profound negative impact on performance. [10, 14, 7]. For example, in a tracking application a target may undergo a sudden change in illumination or may be partially occluded by another target. If these rare events are ignored the tracking algorithm will fail.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 594–607, 2010.

Several alternatives to the Gaussian distribution have been proposed in order to avoid this issue. One such alternative is the multivariate t-distribution [7]. The t-distribution has a similar shape as the Gaussian distribution but with much heavier tails. Because of the heavy tails, the t-distribution is a better model for situations in which rare events commonly occur. The t-distribution is particularly better suited for high-dimensional spaces where all events are expected to be rare. The heavy tails of the t-distribution also increase the robustness in parameter estimation, since the outliers in the data naturally have little overall impact on the parameters [5]. This is in stark contrast to the Gaussian for which a few outliers can dramatically change the parameter estimates of the distribution.

Despite these attractive properties of the t-distribution, it has not been widely used. We believe this can be attributed to the lack of good estimation techniques (in an engineering sense) for the parameters of the distribution. Numerous EM-based iterative algorithms have been developed to compute the maximum likelihood estimates for the parameters of the t-distribution [8, 9, 10]. However, because of their iterative nature, these algorithms are computationally expensive. Also, these algorithms work on the dataset as a whole and cannot be incrementally updated as new data becomes available. This deficiency severely limits their usefulness in real time applications.

This paper addresses the problem of parameter estimation for the multivariate t-distribution. We propose a new approximate algorithm which is both computationally efficient and incrementally updateable. The proposed algorithm provides comparable estimation accuracy compared to the EM-based algorithms while achieving a significant improvement in the computation time. Using the approximation formula, we then develop an approximate incremental probabilistic PCA (PPCA) for the t-distribution. Previous work has extended the idea of PPCA to the t-distribution [17], but with a focus on extending the EM-based maximum likelihood techniques. As we mentioned, these EM-based iterative estimators are computationally expensive and cannot be updated incrementally, posing severe limitations on the range of applications. We present an approximate incremental approach which has equivalent computational requirements as the incremental PPCA approaches for the Gaussian distribution [16, 11].

2 Multivariate t-Distribution

In this section, we will present some useful properties of the t-distribution, many of which come from the seminal work by Kotz and Nadarajah [6].

2.1 Basic Properties

The pdf of the p-variate t-distribution with ν degrees of freedom is given by

$$f(\mathbf{x}) = \frac{\Gamma((p+\nu)/2)}{\Gamma(\nu/2)(\pi\nu)^{p/2}|\mathrm{S}|^{1/2}} \left[1 + \frac{1}{\nu}(\mathbf{x}-\mathbf{c})^T \mathrm{S}^{-1}(\mathbf{x}-\mathbf{c})\right]^{-(p+\nu)/2} \quad (1)$$

where $\mathbf{c} \in \mathbb{R}^p$ is the location parameter and $\mathrm{S} \in \mathbb{R}^{p\times p}$ is the positive definite scale matrix. Notationally we will write $\mathbf{x} \sim t(\mathbf{c}, \mathrm{S}, \nu)$. The vector $\mathbf{c}$ specifies

the location of the single mode of the distribution. The matrix S specifies the relative width of the central mode along each dimension and also the correlation between dimensions. The degrees of freedom ν controls the heaviness of the tails of the distribution. When $\nu = 1$ we have the Cauchy distribution which has very heavy tails while $\nu = \infty$ gives the Gaussian distribution.

Many applications require the computation of the marginal distribution of one or more random variables for which the joint distribution is known. This is easily done with the multivariate t-distribution by simply partitioning the parameters $\mathbf{c}$ and S, i.e. if $\mathbf{x} \sim t(\mathbf{c}, \mathrm{S}, \nu)$ and we define

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \tag{2}$$

$$\mathbf{c} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} \tag{3}$$

$$\mathrm{S} = \begin{bmatrix} \mathrm{S}_{11} & \mathrm{S}_{12} \\ \mathrm{S}_{21} & \mathrm{S}_{22} \end{bmatrix} \tag{4}$$

then $\mathbf{x}_1 \sim t(\mathbf{c}_1, \mathrm{S}_{11}, \nu)$ and $\mathbf{x}_2 \sim t(\mathbf{c}_2, \mathrm{S}_{22}, \nu)$. The conditional distribution $f(\mathbf{x}_2|\mathbf{x}_1)$ is unfortunately not a t-distribution and does not have a particularly clean form. However, the expectation of $\mathbf{x}_2$ given $\mathbf{x}_1$ does have a nice form

$$E\{\mathbf{x}_2|\mathbf{x}_1\} = \mathrm{S}_{21}\mathrm{S}_{11}^{-1}(\mathbf{x}_1 - \mathbf{c}_1) + \mathbf{c}_2 \tag{5}$$

2.2 Sampling from the Multivariate t-Distribution

Generating samples from a multivariate t-distribution is fairly straightforward. If $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathrm{I})$ and $\gamma \sim \chi^2(\nu)$ then the random vector

$$\mathbf{x} = \sqrt{\frac{\nu}{\gamma}}\mathrm{T}^T\mathbf{y} + \mathbf{c} \tag{6}$$

is distributed as $\mathbf{x} \sim t(\mathbf{c}, \mathrm{T}^T\mathrm{T}, \nu)$. Note that every entry in the random vector $\mathbf{x}$ is scaled according to the same value γ. Because of this, even if the scale matrix is diagonal the entries in $\mathbf{x}$ will not be independent. This is an important limitation of the multivariate t-distribution.

3 Batch Parameter Estimation

3.1 Maximum Likelihood Estimator

The maximum likelihood estimates for the parameters of the t-distribution based on sample data $\mathrm{X} = [\mathbf{x}_1\, \mathbf{x}_2\, \ldots\, \mathbf{x}_n]$ must satisfy the following equations [10]

$$\mathbf{c} = \frac{\sum_{i=1}^{n} w_i \mathbf{x}_i}{\sum_{i=1}^{n} w_i} \tag{7}$$

$$\mathrm{S} = \frac{1}{n}\sum_{i=1}^{n} w_i \left(\mathbf{x}_i - \mathbf{c}\right)\left(\mathbf{x}_i - \mathbf{c}\right)^T \tag{8}$$

where

$$w_i = (p+\nu)\left(\nu + (\mathbf{x}_i - \mathbf{c})^T \mathrm{S}^{-1}(\mathbf{x}_i - \mathbf{c})\right)^{-1} \tag{9}$$

These equations cannot be solved to give closed form estimates for the parameters. An EM-based approach can be used to iteratively estimate $\mathbf{c}$, S, and ν which satisfy these constraints [8, 9, 12]. While some variations of the implementation may achieve a faster parameter estimation than others, fundamentally they are all iterative algorithms, thus computationally expensive. More importantly, none of these methods can be extended to efficiently update the estimates as new data becomes available. All of the algorithms are based on computing weighted means and covariances. Since the weight for each sample is a function of $\mathbf{c}$, S, and ν, the weights on old data will change as new data becomes available and hence the old data must be included in the computation.

3.2 Approximate Algorithm

Special Case. To develop an approximate algorithm for computing the parameters we begin by considering the special case $\mathbf{x} \sim t(\mathbf{0}, \alpha\mathrm{I}, \nu)$ for some constant $\alpha > 0$. In this special case the pdf of the norm of $\mathbf{x}$ is given by

$$f(\|\mathbf{x}\|) = \frac{2\|\mathbf{x}\|^{p-1}}{\mathrm{B}(\nu/2, p/2)(\alpha\nu)^{p/2}} \left(1 + \frac{1}{\alpha\nu}\|\mathbf{x}\|^2\right)^{-(\nu+p)/2} \tag{10}$$

where $\mathrm{B}(x,y) = \Gamma(x)\Gamma(y)\Gamma^{-1}(x+y)$ is the beta function. The goal is to estimate ν and α given sample data $\mathrm{X} = [\mathbf{x}_1\, \mathbf{x}_2\, \ldots\, \mathbf{x}_n]$. This can be done by considering the following results

$$E\{\log\|\mathbf{x}\|^2\} = \log\alpha + \log\nu + \psi_0\left(\frac{p}{2}\right) - \psi_0\left(\frac{\nu}{2}\right) \tag{11}$$

$$Var\{\log\|\mathbf{x}\|^2\} = \psi_1\left(\frac{\nu}{2}\right) + \psi_1\left(\frac{p}{2}\right) \tag{12}$$

where $\psi_0(x)$ is the digamma function and $\psi_1(x)$ is the trigamma function.

Let $z_i = \log\|\mathbf{x}_i\|^2 = \log\mathbf{x}_i^T\mathbf{x}_i$. To estimate ν we need to solve for $\hat{\nu}$ which satisfies

$$\psi_1\left(\frac{\hat{\nu}}{2}\right) = \frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z})^2 - \psi_1\left(\frac{p}{2}\right) \tag{13}$$

where $\bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i$. Unfortunately we cannot directly solve Eq. (13). However, by using the approximation

$$\psi_1(x) \approx \frac{x+1}{x^2} \tag{14}$$

we can compute the estimate

$$\hat{\nu} = \frac{1+\sqrt{1+4b}}{b} \tag{15}$$

with

$$b = \frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z})^2 - \psi_1\left(\frac{p}{2}\right) \tag{16}$$

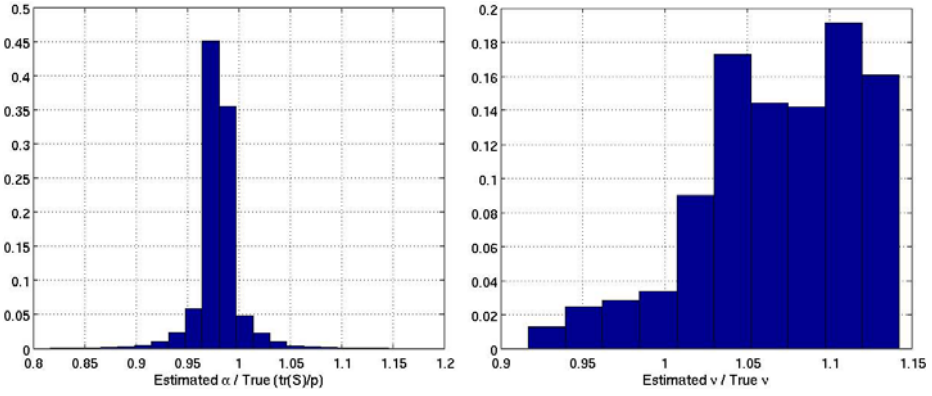


Fig. 1. An experimental evaluation of Eqs. 15 and 17 when applied to samples from general t-distributions. Each figure is a normalized histogram over 10000 trials. For each trial we set $\nu = 10^u$ where $u \sim U(-1, 1)$ is a uniform random variable. The scale matrix for each trial was a random positive definite matrix drawn from a Wishart distribution and p was set to 50. The left figure compares $\hat{\nu}$ computed using Eq. 15 to the true value ν. The right figure compares $\hat{\alpha}$ computed using Eq. 17 to the mean of the diagonal entries of the scale matrix.

Finally, we use Eq. (11) to compute an estimate for the scaling

$$\hat{\alpha} = \exp\left\{\bar{z} - \log\hat{\nu} + \psi_0\left(\frac{\hat{\nu}}{2}\right) - \psi_0\left(\frac{p}{2}\right)\right\}. \tag{17}$$

General Case. We now consider the general case when $\mathbf{x} \sim t(\mathbf{c}, \mathrm{S}, \nu)$. The location vector $\mathbf{c}$ can be estimated by considering each dimension of the data separately and computing either the sample median or the mean of the center 25% of the data [13]. We will use $\hat{\mathbf{c}}$ to denote the estimate of the location vector.

Since our goal is a computationally efficient approximation rather than an exact solution to the parameters we begin by estimating ν and α using the equations of the preceding section, i. e. we assume for the purpose of approximation that $\mathrm{S} = \alpha\mathrm{I}$ for some α. This can be done by first computing $z_i = \log\|\mathbf{x}_i - \hat{\mathbf{c}}\|^2$ and then directly applying Eqs. (15) and (17). In practice, the estimate $\hat{\nu}$ is a good approximation to ν regardless of the structure of S as is shown in Fig. 1. The slight positive bias may be due to the error in the approximation for the trigamma function given in Eq. 14. The scaling estimate $\hat{\alpha}$ also provides a good estimate for the mean of the diagonal entries of S, as illustrated by the results shown in Fig. 1. Hence all that remains is to estimate the relative scaling of the elements of S.

To estimate the relative scaling of the elements of S we use the auxiliary matrix

$$\bar{\mathrm{S}} = \frac{1}{n}\sum_{i=1}^{n}\frac{(\mathbf{x}_i - \hat{\mathbf{c}})(\mathbf{x}_i - \hat{\mathbf{c}})^T}{\|\mathbf{x}_i - \hat{\mathbf{c}}\|^\beta}. \tag{18}$$

which is similar to the sample covariance except that each sample is first scaled by the norm raised to a constant power β. We have experimentally validated that a good choice for β can be given by

$$\beta = \frac{2 \log_2 p}{\hat{\nu}^2 + \log_2 p} \tag{19}$$

Note that for many applications p is large and $\hat{\nu}$ is small so we can directly use $\beta = 2$. The scaling term in the denominator of Eq. 18 is necessary in order to give a good approximation when ν is small. We can now apply the estimated mean of the diagonal entries $\hat{\alpha}$ to obtain an estimate for S

$$\hat{\mathrm{S}} = \frac{\hat{\alpha} p}{\mathrm{tr}\left(\bar{\mathrm{S}}\right)} \bar{\mathrm{S}} \tag{20}$$

This completes the development of the approximation algorithm which is given in succinct form in Fig. 2.

$$\hat{\mathbf{c}} = \text{median of each dimension of the data}$$

$$z_i = \log \|\mathbf{x}_i - \hat{\mathbf{c}}\|^2 \quad \bar{z} = \frac{1}{n} \sum_{i=1}^{n} z_i$$

$$b = \frac{1}{n} \sum_{i=1}^{n} (z_i - \bar{z})^2 - \psi_1 \left(\frac{p}{2}\right)$$

$$\hat{\nu} = \frac{1 + \sqrt{1 + 4b}}{b}$$

$$\hat{\alpha} = \exp\left\{ \bar{z} - \log \hat{\nu} + \psi_0 \left(\frac{\hat{\nu}}{2}\right) - \psi_0 \left(\frac{p}{2}\right) \right\}$$

$$\beta = \frac{2 \log_2 p}{\hat{\nu}^2 + \log_2 p}$$

$$\bar{\mathrm{S}} = \frac{1}{n} \sum_{i=1}^{n} \frac{(\mathbf{x}_i - \hat{\mathbf{c}})(\mathbf{x}_i - \hat{\mathbf{c}})^T}{\|\mathbf{x}_i - \hat{\mathbf{c}}\|^{\beta}}.$$

$$\hat{\mathrm{S}} = \frac{\hat{\alpha} p}{\mathrm{tr}\left(\bar{\mathrm{S}}\right)} \bar{\mathrm{S}}$$

Fig. 2. Batch Approximation Algorithm

3.3 Comparative Evaluation of Maximum Likelihood and Approximation Algorithms

To evaluate the accuracy of the approximation algorithm we performed several experiments on synthetic data. In three experiments we varied separately the

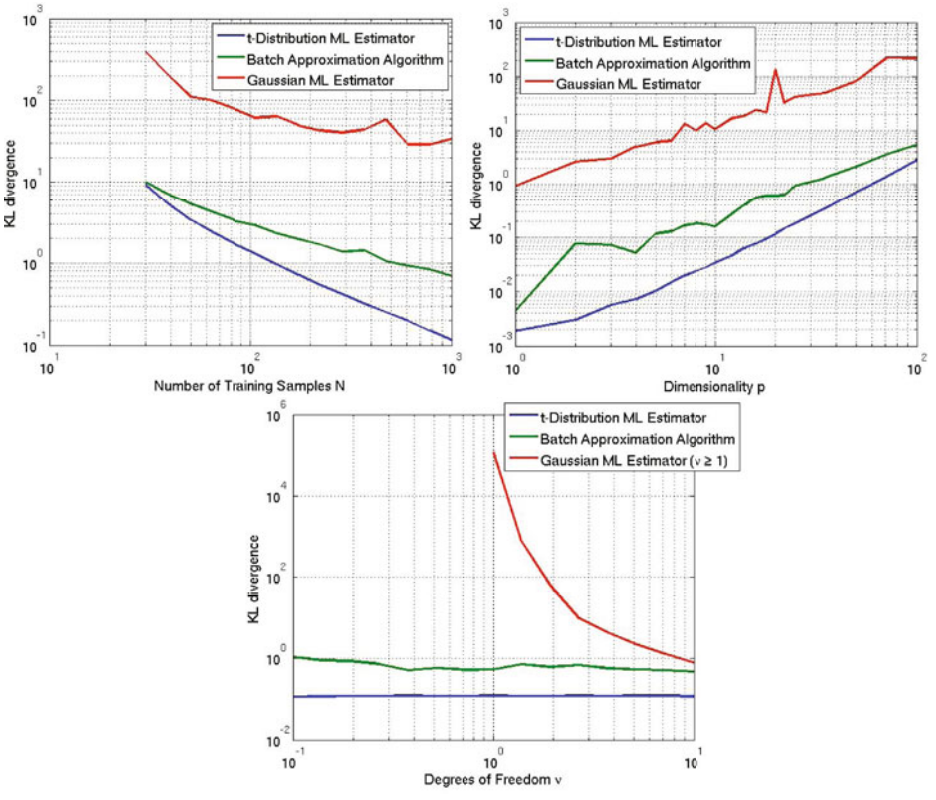


Fig. 3. Comparison of the accuracy of the maximum likelihood and approximation algorithms for estimating the parameters of a multivariate t-distribution. The accuracy of the maximum likelihood Gaussian distribution is provided for comparison.

dimensionality p, the degree of freedom ν, and the number of training samples N. In each case we generated synthetic data using the sampling technique described in section 2.2 and then computed the KL divergence from the true distribution for both the maximum likelihood parameter estimates (computed using the method in [9]) and the approximate parameter estimates. We also computed the KL divergence from the true distribution for the maximum likelihood Gaussian distribution in order to give a basis for comparison. The results are shown in Fig. 3. As expected, the KL divergence of the approximation algorithm is higher than that of the maximum likelihood algorithm. However, the approximation algorithm is nearly as good across a broad range of parameter settings and in particular it is significantly better than the maximum likelihood Gaussian in every case.

The maximum likelihood estimator has slightly better accuracy but in many other ways the approximate algorithm is superior. The primary advantages of the approximate algorithm are

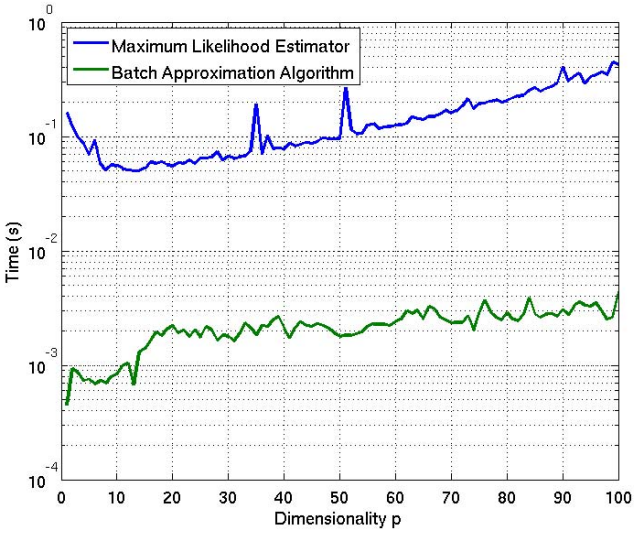


Fig. 4. Training time on 200 samples as a function of p for the maximum likelihood and approximate algorithms

- **Computational Efficiency:** Fig. 4 shows the running time for both methods as a function of the dimensionality p of the data based on a MATLAB implementation. The approximate algorithm is consistently 50-100 times faster.
- **Easy Implementation:** Because the approximate algorithm is directly computed there is no need for iterative looping in the code. This also eliminates the need to check for convergence.
- **Useful Theoretical Tool:** We can use the approximate parameter estimation equations as a basis for developing additional algorithms which would not be possible with the maximum likelihood estimator, e.g. incremental algorithms.

4 Incremental Parameter Estimation

Many real-time applications require online updating of the parameters of the distribution. To handle this situation we present two incremental approaches which can be used with the t-distribution based on the batch approximation algorithm of the preceding section. The first approach is essentially a direct extension of the batch algorithm. The second approach uses PPCA to estimate the parameters under the assumption that the underlying dimensionality of the model is much lower than the true dimensionality. Note that for both algorithms, we can incrementally estimate $\hat{\mathbf{c}}$ without needing to store previously seen data by using an online quantile estimator [15, 2].

4.1 Direct Incremental Algorithm

In order to convert the batch algorithm to an incremental algorithm we need to rewrite Eqs. (15), (17), and (18) to be incremental. To compute $\hat{\nu}$ and $\hat{\alpha}$ we need to incrementally update estimates for the mean and variance of $z = \log \|\mathbf{x} - \hat{\mathbf{c}}\|^2$. After the kth sample, the mean $\bar{z}$ and variance v_z are updated by

$$\bar{z}^{(k)} = \frac{k-1}{k}\bar{z}^{(k-1)} + \frac{1}{k}z_k \tag{21}$$

$$v_z^{(k)} = \frac{k-1}{k}v_z^{(k-1)} + \frac{k-1}{k^2}\left(z_k - \bar{z}^{(k-1)}\right)^2 \tag{22}$$

where $z_k = \log \|\mathbf{x}_k - \hat{\mathbf{c}}^{(k)}\|^2$, i.e. we use the best available estimate for $\mathbf{c}$ for each incremental update. Because the estimate for $\mathbf{c}$ changes with each sample these incremental update formulas will not give exactly the same results as the batch algorithm. In practice this is typically not a problem. However, when k is very small we must be careful to ensure that $\|\mathbf{x}_k - \hat{\mathbf{c}}^{(k)}\|^2 \neq 0$. One way to do this is to store the first few samples and use these to compute a batch estimate before switching to the incremental estimator.

We can now directly use Eq. (15) to conclude that the estimate for ν after the kth sample is given by

$$\hat{\nu}^{(k)} = \frac{1 + \sqrt{1 + 4b^{(k)}}}{b^{(k)}} \tag{23}$$

where

$$b^{(k)} = v_z^{(k)} - \psi_1\left(\frac{p}{2}\right) \tag{24}$$

Similarly, the estimate for α is given by

$$\hat{\alpha}^{(k)} = \exp\left\{\bar{z}^{(k)} - \log\hat{\nu}^{(k)} + \psi_0\left(\frac{\hat{\nu}^{(k)}}{2}\right) - \psi_0\left(\frac{p}{2}\right)\right\}. \tag{25}$$

The last step is to compute an estimate for $\bar{\mathrm{S}}$. Under the assumption that p is large and ν is small (and hence $\beta = 2$ in Eq. (19)) we use the estimate

$$\bar{\mathrm{S}}^{(k)} = \frac{k-1}{k}\bar{\mathrm{S}}^{(k-1)} + \frac{1}{k}\left[\frac{(\mathbf{x}_k - \hat{\mathbf{c}}^{(k)})(\mathbf{x}_k - \hat{\mathbf{c}}^{(k)})^T}{\|\mathbf{x}_k - \hat{\mathbf{c}}^{(k)}\|^2}\right] \tag{26}$$

where again we use the best available estimate for $\mathbf{c}$ for each update. Again we must be careful to ensure that $\|\mathbf{x}_k - \hat{\mathbf{c}}^{(k)}\|^2 \neq 0$. This is most likely to occur when k is very small and as a solution, as already stated, we use the first few samples to compute a batch estimate of $\bar{\mathrm{S}}$ before switching to the incremental algorithm.

4.2 PPCA for t-Distribution

Although PPCA was originally developed in the context of the multivariate Gaussian distribution the idea has been extended to the t-distribution [16, 17]. The idea behind PPCA is to model the scale matrix in the following way

$$\mathrm{S} = s\mathrm{I} + \mathrm{W}\mathrm{W}^T \tag{27}$$

where $s > 0$ captures the general level of uncertainty in the random variable while $\mathrm{W} \in \mathbb{R}^{p \times q}$, $q < p$, captures the correlation between dimensions. Since we typically have $q \ll p$, this model for S can be trained with significantly fewer data samples while still providing a powerful model.

The maximum likelihood estimates for W and s can be obtained through an iterative EM-based approach [17]. Once again, this approach is too slow for practical use in many computer vision problems. As an alternative, we present an incremental algorithm based on the approximate incremental estimator of the preceding section. The key is to note that the incremental equation for $\hat{\alpha}$ given in the preceding section is still applicable and so instead of directly modeling S as in Eq. 27 we can instead model $\bar{\mathrm{S}}$. Specifically, the goal is to find estimates for $\hat{s}$ and $\hat{\mathrm{W}}$ such that

$$\bar{\mathrm{S}}^{(k)} \approx \hat{s}^{(k)}\mathrm{I} + \hat{\mathrm{W}}^{(k)}\left(\hat{\mathrm{W}}^{(k)}\right)^T \tag{28}$$

Since $\bar{\mathrm{S}}$ is in essence a weighted covariance matrix the incremental update formulas for PPCA with the multivariate Gaussian distribution can be used as a template for how to estimate $\hat{s}$ and $\hat{\mathrm{W}}$ [11]. The idea is to use

$$\hat{\mathrm{W}}^{(k)} = \mathrm{V}^{(k)}(\Lambda^{(k)} - \hat{s}^{(k)}\mathrm{I})^{1/2} \tag{29}$$

where $\Lambda^{(k)}$ is a diagonal matrix of the q largest eigenvalues of $\bar{\mathrm{S}}^{(k)}$ and the columns of $\mathrm{V}^{(k)} \in \mathbb{R}^{p \times q}$ are the corresponding eigenvectors.

The first step is to rewrite the incremental update equation for $\bar{\mathrm{S}}$ as

$$\bar{\mathrm{S}}^{(k)} = \frac{k-1}{k}\left(\bar{\mathrm{S}}^{(k-1)} + \mathbf{y}\mathbf{y}^T\right) \tag{30}$$

where

$$\mathbf{y} = \frac{1}{\sqrt{k-1}} \frac{\mathbf{x}_k - \hat{\mathbf{c}}^{(k)}}{\|\mathbf{x}_k - \hat{\mathbf{c}}^{(k)}\|} \tag{31}$$

Let $\mathrm{L} = \left[\hat{\mathrm{W}}^{(k-1)}\,\mathbf{y}\right]$ and let $\mathrm{Q} = \mathrm{L}^T\mathrm{L}$. Compute an eigen decomposition of $\mathrm{Q} \in \mathbb{R}^{q \times q}$ s.t.

$$\mathrm{Q} = \mathrm{U}\Gamma\mathrm{U}^T \tag{32}$$

where $\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_{q+1})$. Then the first $q+1$ eigenvalues of $\bar{\mathrm{S}}^{(k)}$ are given by

$$\lambda_i = \frac{n}{n+1}[\hat{s}^{(k-1)} + \gamma_i] \tag{33}$$

and the corresponding eigenvectors are given by the columns of

$$\hat{\mathrm{V}} = \mathrm{L}\mathrm{U}\Gamma^{-1/2} \tag{34}$$

Note that we keep only the first q eigenvalues and eigenvectors in order to compute $\hat{\mathrm{W}}^{(k)}$. Finally, we update $\hat{s}$

$$\hat{s}^{(k)} = \frac{n}{n+1}\left[\frac{\gamma_{q+1}}{p-q} + \hat{s}^{(k-1)}\right] \tag{35}$$

5 Application to Computer Vision

5.1 Classification

A common task in computer vision is to determine which object from a set of possible choices is visible in a small subsection of the image. One way to solve this problem is to first train a probability model for each possible choice based on training data. The best estimate for which object is visible in a small subsection of the image is then given by the probability model which assigns the highest probability to the subsection. This method of classification is known as the generative approach.

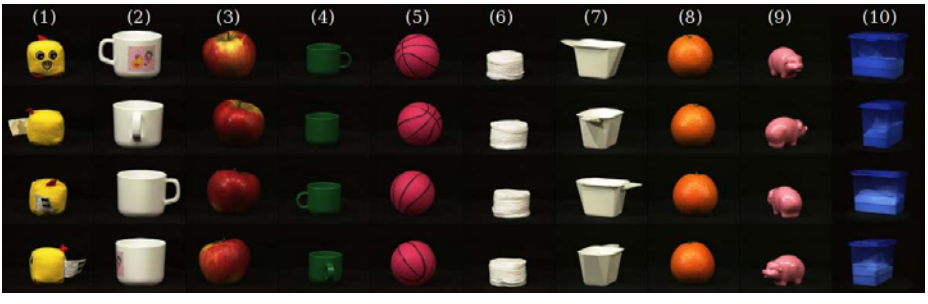


Fig. 5. Objects from the Amsterdam Library of Object Images (ALOI)[3]

In order to compare the power of the Gaussian and t-distributions for solving the classification problem, we analyzed ten objects (shown in Fig. 5) from the Amsterdam Library of Object Images [3]. For each object, there are 72 images taken in 5° increments around the object. We randomly split these images into 36 training images and 36 testing images for each object. For each image, we then extracted the brightness of the pixels from 100 non-overlapping 10×10 squares and used these as the data samples. The data samples from the training images were used to obtain the maximum likelihood Gaussian distribution and the approximate t-distribution using the proposed batch algorithm. The probability models that had been learned for all of the objects were then used to classify the samples from the testing images.

Under these conditions, the Gaussian distribution led to a classification accuracy of 51% while using the t-distribution significantly improved the accuracy to 68%. The reason for this can be seen by considering Table 1 which gives individual results for each object. The Gaussian distribution gives very poor results for objects 2, 8, and 9; each of which has substantial changes in brightness due to the design, specular highlights, and shadows. These changes represent outliers and are poorly handled by the Gaussian model, resulting in a very broad distribution with poor discrimination. Objects 4 and 6 on the other hand, which give good results with a Gaussian distribution, are mostly uniform in brightness and do not undergo significant changes from frame to frame.

Table 1. Object classification rates in %. Each entry gives the percentage of samples that were correctly classified for that object.

	1	2	3	4	5	6	7	8	9	10
t-distribution	74	66	66	90	61	97	47	62	43	71
Gaussian	74	25	44	91	43	80	63	24	8	55

The parameter estimation algorithm for the t-distribution automatically includes robustness against outliers and so large changes in brightness have little effect on the overall parameter estimation. The result is a tighter distribution compared to the Gaussian. Because of this the t-distribution more effectively models each object and hence gives better discrimination. Note that the algorithm also performs very well when no outliers are present, giving excellent results for objects 4 and 6. It is this flexibility to handle a wide range of data types which makes the t-distribution an ideal choice for many applications.

5.2 Tracking

Tracking is another very important application in computer vision. The goal in tracking is to identify which pixels in each frame of a video sequence were generated by one or more targets. This can be done by training a probability distribution over the brightness of the pixels making up each target. The joint

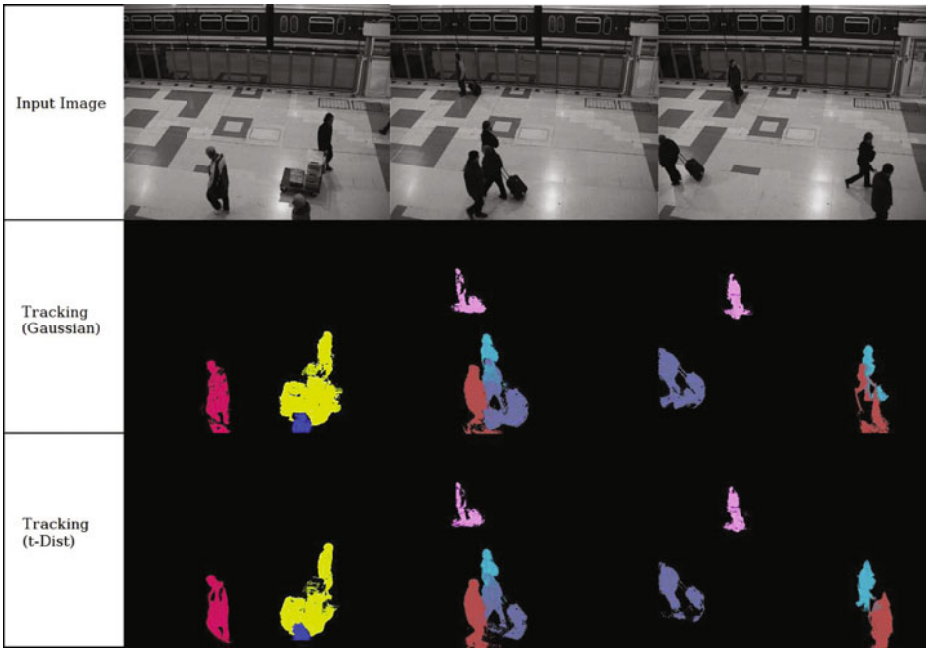


Fig. 6. Tracking results using the Gaussian distribution and the t-distribution

distribution is used to identify where a target is located in a given frame. The marginal distributions can then be used to determine for each pixel if it was generated by the target or something else, effectively segmenting out the target from its surroundings.

Using a tracking algorithm based on PPCA for the Gaussian distribution as a basis we modified the algorithm to use the t-distribution instead [1]. Both algorithms were tested on a video sequence from PETS2006 [4]. The results for three frames of the video sequence are shown in Fig. 6. The complete video sequence is included with the supplementary material. Although the overall results are similar regardless of which distribution is used, the t-distribution does show improved performance. The t-distribution is much less susceptible to shadows which can be seen by looking at the gray target in the second and third frames. The t-distribution also handles overlapping targets more cleanly. Because of this it is able to properly distinguish between the orange and cyan targets in the final frame while the Gaussian distribution confuses them.

6 Conclusions

The Gaussian distribution is by far the most commonly used parametric probability model mainly because it is simple to use and computationally tractable even for high dimensional data. The light tails of the Gaussian distribution, however, make it a poor model for the randomness present in many sources of data. We believe the t-distribution represents a viable replacement for the Gaussian. By developing an approximate algorithm to compute the parameters, we have shown that the t-distribution can be made as computationally efficient as the Gaussian. Furthermore, we show that the proposed algorithm can be updated online for real time applications. Even though the parameter estimation is only approximate, the results show that the t-distribution outperforms the Gaussian for two important applications in computer vision. We expect future research along these lines to touch a large spectrum of domains in computer vision.

Acknowledgment

The authors want to thank Sierra Nevada Corporation for their support.

References

[1] Aeschliman, C., Park, J., Kak, A.C.: A Probabilistic Framework for Joint Segmentation and Tracking. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2010)

[2] Chen, F., Lambert, D., Pinheiro, J.C.: Incremental quantile estimation for massive tracking. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 516–522. ACM, New York (2000)

[3] Geusebroek, J.M., Burghouts, G.J., Smeulders, A.W.M.: The Amsterdam library of object images. International Journal of Computer Vision 61(1), 103–112 (2005)

[4] Iscaps, C.: Pets2006 (2006), http://www.cvg.rdg.ac.uk/pets2006/data.html
[5] Khan, Z., Balch, T., Dellaert, F.: MCMC-based particle filtering for tracking a variable number of interacting targets. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1805–1918 (2005)
[6] Kotz, S., Nadarajah, S.: Multivariate t distributions and their applications. Cambridge Univ. Pr., Cambridge (2004)
[7] Lange, K.L., Little, R.J.A., Taylor, J.M.G.: Robust statistical modeling using the t distribution. Journal of the American Statistical Association, 881–896 (1989)
[8] Liu, C., Rubin, D.B.: ML estimation of the t distribution using EM and its extensions, ECM and ECME. Statistica Sinica 5(1), 19–39 (1995)
[9] Meng, X.L., van Dyk, D.: The EM algorithm–an old folk-song sung to a fast new tune. Journal of the Royal Statistical Society. Series B (Methodological), 511–567 (1997)
[10] Nadarajah, S., Kotz, S.: Estimation Methods for the Multivariate t Distribution. Acta Applicandae Mathematicae: An International Survey Journal on Applying Mathematics and Mathematical Applications 102(1), 99–118 (2008)
[11] Nguyen, H.T., Ji, Q., Smeulders, A.W.M.: Spatio-temporal context for robust multitarget tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(1), 52 (2007)
[12] Peel, D., McLachlan, G.: Robust mixture modelling using the t distribution. Statistics and Computing 10(4), 339–348 (2000)
[13] Rothenberg, T.J., Fisher, F.M., Tilanus, C.B.: A note on estimation from a Cauchy sample. Journal of the American Statistical Association 59(306), 460–463 (1964)
[14] Simoncelli, E.P.: Statistical modeling of photographic images. In: Handbook of Image and Video Processing, pp. 431–441 (2005)
[15] Tierney, L.: A space-efficient recursive procedure for estimating a quantile of an unknown distribution. SIAM Journal on Scientific and Statistical Computing 4, 706 (1983)
[16] Tipping, M., Bishop, C.M.: Probabilistic principal component analysis. Journal of the Royal Statistical Society. Series B (Statistical Methodology) 61(3), 611–622 (1999)
[17] Zhao, J., Jiang, Q.: Probabilistic PCA for t distributions. Neurocomputing 69(16-18), 2217–2226 (2006)

LACBoost and FisherBoost: Optimally Building Cascade Classifiers

Chunhua Shen[1,2], Peng Wang[3,⋆], and Hanxi Li[2,1]

[1] NICTA⋆⋆, Canberra Research Laboratory, ACT 2601, Australia
[2] Australian National University, ACT 0200, Australia
[3] Beihang University, Beijing 100191, China

Abstract. Object detection is one of the key tasks in computer vision. The cascade framework of Viola and Jones has become the *de facto* standard. A classifier in each node of the cascade is required to achieve extremely high detection rates, instead of low overall classification error. Although there are a few reported methods addressing this requirement in the context of object detection, there is no a principled feature selection method that explicitly takes into account this asymmetric node learning objective. We provide such a boosting algorithm in this work. It is inspired by the linear asymmetric classifier (LAC) of [1] in that our boosting algorithm optimizes a similar cost function. The new totally-corrective boosting algorithm is implemented by the column generation technique in convex optimization. Experimental results on face detection suggest that our proposed boosting algorithms can improve the state-of-the-art methods in detection performance.

1 Introduction

Real-time object detection has been extensively studied in the past a few years due to its important applications in surveillance, intelligent video analysis *etc.* Viola and Jones proffered the first real-time face detector [2,3]. To date, it is still considered one of the state-of-the-art, and their framework is the basis of many incremental work afterwards. Object detection is a highly asymmetric classification problem with the exhaustive scanning-window search being used to locate the target in an image. Only a few are true target objects among the millions of scanned patches. Cascade classifiers have been proposed for efficient detection, which takes the asymmetric structure into consideration. Under the assumption of each node of the cascade classifier makes independent classification errors, the detection rate and false positive rate of the entire cascade are: $F_{\text{dr}} = \prod_{t=1}^{N} d_t$ and $F_{\text{fp}} = \prod_{t=1}^{N} f_t$, respectively. As pointed out in [2,1], these two

⋆ P. Wang's contribution was made when visiting NICTA and Australian National University.

⋆⋆ NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 608–621, 2010.

equations suggest a *node learning objective*: Each node should have an extremely high detection rate d_t (*e.g.*, 99.7%) and a moderate false positive rate f_t (*e.g.*, 50%). With the above values of d_t and f_t, assume that the cascade has $N = 20$ nodes, then $F_{\text{dr}} \approx 94\%$ and $F_{\text{fp}} \approx 10^{-6}$, which is a reasonable design goal.

A drawback of standard boosting like AdaBoost is that it does not take advantage of the cascade classifier. AdaBoost only minimizes the overall classification error and does not minimize the number of false negatives. In this sense, the features selected are not optimal for the purpose of rejecting negative examples. At the feature selection and classifier training level, Viola and Jones leveraged the asymmetry property, to some extent, by replacing AdaBoost with AsymBoost [3]. AsymBoost incurs more loss for misclassifying a positive example by simply modifying AdaBoost's exponential loss. Better detection rates were observed over the standard AdaBoost. Nevertheless, AsymBoost addresses the node learning goal *indirectly* and still may not be the optimal solution. Wu *et al.* explicitly studied the node learning goal and they proposed to use linear asymmetric classifier (LAC) and Fisher linear discriminant analysis (LDA) to adjust the linear coefficients of the selected weak classifiers [1,4]. Their experiments indicated that with this post-processing technique, the node learning objective can be better met, which is translated into improved detection rates. In Viola and Jones' framework, boosting is used to select features and at the same time to train a strong classifier. Wu *et al.*'s work separates these two tasks: they still use AdaBoost or AsymBoost to select features; and at the second step, they build a strong classifier using LAC or LDA. Since there are two steps here, in Wu *et al.*'s work [1,4], the node learning objective is only considered at the second step. At the first step—feature selection—the node learning objective is not explicitly considered. We conjecture that *further improvement may be gained if the node learning objective is explicitly taken into account at both steps*. We design new boosting algorithms to implement this idea and verify this conjecture. Our major contributions are as follows.

1. We develop new boosting-like algorithms by directly minimizing the objective function of linear asymmetric classifier, which is termed as LACBoost (and FisherBoost from Fisher LDA). Both of them can be used to select features that is optimal for achieving the node learning goal in training a cascade classifier. To our knowledge, this is the first attempt to design such a feature selection method.
2. LACBoost and FisherBoost share similarities with LPBoost [5] in the sense that both use column generation—a technique originally proposed for large-scale linear programming (LP). Typically, the Lagrange dual problem is solved at each iteration in column generation. We instead solve the primal quadratic programming (QP) problem, which has a special structure and entropic gradient (EG) can be used to solve the problem very efficiently. Compared with general interior-point based QP solvers, EG is much faster. Considering one needs to solve QP problems a few thousand times for training a complete cascade detector, the efficiency improvement is enormous. Compared with training an AdaBoost based cascade detector, the

time needed for LACBoost (or FisherBoost) is comparable. This is because for both cases, the majority of the time is spent on weak classifier training and bootstrapping.

3. We apply LACBoost and FisherBoost to face detection and better performances are observed over the state-of-the-art methods [1,4]. The results confirm our conjecture and show the effectiveness of LACBoost and FisherBoost. LACBoost can be immediately applied to other asymmetric classification problems.
4. We also analyze the condition that makes the validity of LAC, and show that the multi-exit cascade might be more suitable for applying LAC learning of [1,4] (and our LACBoost) rather than Viola-Jones standard cascade.

Besides these, the LACBoost/FisherBoost algorithm differs from traditional boosting algorithms in that LACBoost/FisherBoost does not minimize a loss function. This opens new possibilities for designing new boosting algorithms for special purposes. We have also extended column generation for optimizing nonlinear optimization problems.

Related work. There are three important components that make Viola and Jones' framework tremendously successful [2]: (1) The cascade classifier that efficiently filters out most negative patches in early nodes; and also contributes to enable the final classifier to have a very high detection rate; (2) AdaBoost that selects informative features and at the same time trains a strong classifier; (3) The use of integral images, which makes the computation of Haar features extremely fast. Most of the work later improves one or more of these three components. In terms of the cascade classifier, a few different approaches such as soft cascade [6], dynamic cascade [7], and multi-exit cascade [8]. We have used the multi-exit cascade in this work. The multi-exit cascade tries to improve the classification performance by using all the selected weak classifiers for each node. So for the n-th strong classifier (node), it uses all the weak classifiers in this node as well as those in the previous $n-1$ nodes. We show that the LAC post-processing can enhance the multi-exit cascade. More importantly, we show that the multi-exit cascade better meets LAC's requirement of data being Gaussian distributions. The second research topic is the learning algorithm for constructing a classifier. Wu *et al.* proposed LAC to learn a better strong classifier [1]. Li *et al.* advocated FloatBoost to discard some redundant weak classifiers during AdaBoost's greedy selection procedure [9]. Liu and Shum proposed KLBoost to select features and train a strong classifier [10]. Other variants of boosting have been applied to detection.

Notation. The following notation is used. A matrix is denoted by a bold upper-case letter ($\mathbf{X}$); a column vector is denoted by a bold lower-case letter ($\mathbf{x}$). The ith row of $\mathbf{X}$ is denoted by $\mathbf{X}_{i:}$ and the i-th column $\mathbf{X}_{:i}$. The identity matrix is $\mathbf{I}$ and its size should be clear from the context. $\mathbf{1}$ and $\mathbf{0}$ are column vectors of 1's and 0's, respectively. We use $\succcurlyeq, \preccurlyeq$ to denote component-wise inequalities.

Let $\{(\mathbf{x}_i, y_i)\}_{i=1,\cdots,m}$ be the set of training data, where $\mathbf{x}_i \in \mathcal{X}$ and $y_i \in \{-1, +1\}$, $\forall i$. The training set consists of m_1 positive training points and m_2

negative ones; $m_1 + m_2 = m$. Let $h(\cdot) \in \mathcal{H}$ be a weak classifier that projects an input vector $\mathbf{x}$ into $\{-1, +1\}$. Here we only consider discrete classifier outputs. We assume that the set $\mathcal{H}$ is finite and we have n possible weak classifiers. Let the matrix $\mathbf{H} \in \mathbb{R}^{m \times n}$ where the (i, j) entry of $\mathbf{H}$ is $\mathbf{H}_{ij} = h_j(\mathbf{x}_i)$. $\mathbf{H}_{ij}$ is the label predicted by weak classifier $h_j(\cdot)$ on the training datum $\mathbf{x}_i$. We define a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ such that its (i, j) entry is $\mathbf{A}_{ij} = y_i h_j(\mathbf{x}_i)$.

2 Linear Asymmetric Classification

Before we propose our LACBoost and FisherBoost, we briefly overview the concept of LAC. Wu *et al.* [4] have proposed linear asymmetric classification (LAC) as a post-processing step for training nodes in the cascade framework. LAC is guaranteed to get an optimal solution under the assumption of Gaussian data distributions.

Suppose that we have a linear classifier $f(\mathbf{x}) = \mathbf{sign}(\mathbf{w}^\top \mathbf{x} - b)$, if we want to find a pair of $\{\mathbf{w}, b\}$ with a very high accuracy on the positive data $\mathbf{x}_1$ and a moderate accuracy on the negative $\mathbf{x}_2$, which is expressed as the following problem:

$$\max_{\mathbf{w} \neq \mathbf{0}, b} \Pr_{\mathbf{x}_1 \sim (\mu_1, \mathbf{\Sigma}_1)} \{\mathbf{w}^\top \mathbf{x}_1 \geq b\}, \text{ s.t. } \Pr_{\mathbf{x}_2 \sim (\mu_2, \mathbf{\Sigma}_2)} \{\mathbf{w}^\top \mathbf{x}_2 \leq b\} = \lambda, \tag{1}$$

where $\mathbf{x} \sim (\mu, \mathbf{\Sigma})$ denotes a symmetric distribution with mean μ and covariance $\mathbf{\Sigma}$. If we prescribe λ to 0.5 and assume that for any $\mathbf{w}$, $\mathbf{w}^\top \mathbf{x}_1$ is Gaussian and $\mathbf{w}^\top \mathbf{x}_2$ is symmetric, then (1) can be approximated by

$$\max_{\mathbf{w} \neq \mathbf{0}} \frac{\mathbf{w}^\top (\mu_1 - \mu_2)}{\sqrt{\mathbf{w}^\top \mathbf{\Sigma}_1 \mathbf{w}}}. \tag{2}$$

(2) is similar to LDA's optimization problem

$$\max_{\mathbf{w} \neq \mathbf{0}} \frac{\mathbf{w}^\top (\mu_1 - \mu_2)}{\sqrt{\mathbf{w}^\top (\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2) \mathbf{w}}}. \tag{3}$$

(2) can be solved by eigen-decomposition and a close-formed solution can be derived:

$$\mathbf{w}^\star = \mathbf{\Sigma}_1^{-1} (\mu_1 - \mu_2), \quad b^\star = \mathbf{w}^{\star\top} \mu_2. \tag{4}$$

On the other hand, each node in cascaded boosting classifiers has the following form:

$$f(\mathbf{x}) = \mathbf{sign}(\mathbf{w}^\top \mathbf{H}(\mathbf{x}) - b), \tag{5}$$

We override the symbol $\mathbf{H}(\mathbf{x})$ here, which denotes the output vector of all weak classifiers over the datum $\mathbf{x}$. We can cast each node as a linear classifier over the feature space constructed by the binary outputs of all weak classifiers. For each node in cascade classifier, we wish to maximize the detection rate as high as possible, and meanwhile keep the false positive rate to a moderate level (*e.g.*, 50.0%). That is to say, the problem (1) expresses the node learning goal. Therefore, we can

use boosting algorithms (*e.g.*, AdaBoost) as feature selection methods, and then use LAC to learn a linear classifier over those binary features chosen by boosting. The advantage is that LAC considers the asymmetric node learning explicitly.

However, there is a precondition of LAC's validity. That is, for any $\mathbf{w}$, $\mathbf{w}^\top \mathbf{x}_1$ is a Gaussian and $\mathbf{w}^\top \mathbf{x}_2$ is symmetric. In the case of boosting classifiers, $\mathbf{w}^\top \mathbf{x}_1$ and $\mathbf{w}^\top \mathbf{x}_2$ can be expressed as the margin of positive data and negative data. Empirically Wu *et al.* [4] verified that $\mathbf{w}^\top \mathbf{x}$ is Gaussian approximately for a cascade face detector. We discuss this issue in the experiment part in more detail.

3 Constructing Boosting Algorithms from LDA and LAC

In kernel methods, the original data are nonlinearly mapped to a feature space and usually the mapping function $\phi(\cdot)$ is not explicitly available. It works through the inner product of $\phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$. In boosting [11], the mapping function can be seen as explicitly known through: $\phi(\mathbf{x}) : \mathbf{x} \mapsto [h_1(\mathbf{x}), \ldots, h_n(\mathbf{x})]$. Let us consider the Fisher LDA case first because the solution to LDA will generalize to LAC straightforwardly, by looking at the similarity between (2) and (3).

Fisher LDA maximizes the between-class variance and minimizes the within-class variance. In the binary-class case, we can equivalently rewrite (3) into

$$\max_{\mathbf{w}} \ \frac{(\mu_1 - \mu_2)^2}{\sigma_1 + \sigma_2} = \frac{\mathbf{w}^\top \mathbf{C}_b \mathbf{w}}{\mathbf{w}^\top \mathbf{C}_w \mathbf{w}}, \tag{6}$$

where $\mathbf{C}_b$ and $\mathbf{C}_w$ are the between-class and within-class scatter matrices; μ_1 and μ_2 are the projected centers of the two classes. The above problem can be equivalently reformulated as

$$\min_{\mathbf{w}} \ \mathbf{w}^\top \mathbf{C}_w \mathbf{w} - \theta(\mu_1 - \mu_2) \tag{7}$$

for some certain constant θ and under the assumption that $\mu_1 - \mu_2 \geq 0$.[1] Now in the feature space, our data are $\phi(\mathbf{x}_i)$, $i = 1 \ldots m$. We have

$$\mu_1 = \frac{1}{m_1} \mathbf{w}^\top \sum_{y_i=1} \phi(\mathbf{x}_i) = \frac{1}{m_1} \sum_{y_i=1} \mathbf{A}_{i:} \mathbf{w} = \frac{1}{m_1} \sum_{y_i=1} (\mathbf{A}\mathbf{w})_i = \boldsymbol{e}_1^\top \mathbf{A}\mathbf{w}, \tag{8}$$

where $\mathbf{A}_{i:}$ is the i-th row of $\mathbf{A}$.

$$\mu_2 = \frac{1}{m_2} \mathbf{w}^\top \sum_{y_i=-1} \phi(\mathbf{x}_i) = \frac{1}{m_2} \sum_{y_i=-1} \mathbf{H}_{i:} \mathbf{w} = -\boldsymbol{e}_2^\top \mathbf{A}\mathbf{w}, \tag{9}$$

Here the i-th entry of $\boldsymbol{e}_1$ is defined as $\boldsymbol{e}_{1i} = 1/m_1$ if $y_i = +1$, otherwise $\boldsymbol{e}_{1i} = 0$. Similarly $\boldsymbol{e}_{2i} = 1/m_2$ if $y_i = -1$, otherwise $\boldsymbol{e}_{2i} = 0$. We also define $\boldsymbol{e} = \boldsymbol{e}_1 + \boldsymbol{e}_2$. For ease of exposition, we order the training data according to their labels. So the vector $\boldsymbol{e} \in \mathbb{R}^m$:

$$\boldsymbol{e} = [1/m_1, \cdots, 1/m_2, \cdots]^\top, \tag{10}$$

[1] In our face detection experiment, we found that this assumption could always be satisfied.

and the first m_1 components of $\boldsymbol{\rho}$ correspond to the positive training data and the remaining ones correspond to the m_2 negative data. So we have $\mu_1 - \mu_2 = \boldsymbol{e}^\top \boldsymbol{\rho}$, $\mathbf{C}_w = m_1/m \cdot \boldsymbol{\Sigma}_1 + m_2/m \cdot \boldsymbol{\Sigma}_2$ with $\boldsymbol{\Sigma}_{1,2}$ the covariance matrices. By noticing that

$$\mathbf{w}^\top \boldsymbol{\Sigma}_{1,2} \mathbf{w} = \frac{1}{m_{1,2}(m_{1,2}-1)} \sum_{i>k, y_i=y_k=\pm 1} (\rho_i - \rho_k)^2,$$

we can easily rewrite the original problem into:

$$\min_{\mathbf{w},\boldsymbol{\rho}} \tfrac{1}{2}\boldsymbol{\rho}^\top \mathbf{Q}\boldsymbol{\rho} - \theta \boldsymbol{e}^\top \boldsymbol{\rho}, \quad \text{s.t. } \mathbf{w} \succcurlyeq \mathbf{0}, \mathbf{1}^\top \mathbf{w} = 1, \rho_i = (\mathbf{A}\mathbf{w})_i, i = 1, \cdots, m. \tag{11}$$

Here $\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_2 \end{bmatrix}$ is a block matrix with

$$\mathbf{Q}_1 = \begin{bmatrix} \frac{1}{m} & -\frac{1}{m(m_1-1)} & \cdots & -\frac{1}{m(m_1-1)} \\ -\frac{1}{m(m_1-1)} & \frac{1}{m} & \cdots & -\frac{1}{m(m_1-1)} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{m(m_1-1)} & -\frac{1}{m(m_1-1)} & \cdots & \frac{1}{m} \end{bmatrix},$$

and $\mathbf{Q}_2$ is similarly defined by replacing m_1 with m_2 in $\mathbf{Q}_1$. Also note that we have introduced a constant $\frac{1}{2}$ before the quadratic term for convenience. The normalization constraint $\mathbf{1}^\top \mathbf{w} = 1$ removes the scale ambiguity of $\mathbf{w}$. Otherwise the problem is ill-posed.

In the case of LAC, the covariance matrix of the negative data is not involved, which corresponds to the matrix $\mathbf{Q}_2$ is zero. So we can simply set $\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$ and (11) becomes the optimization problem of LAC.

At this stage, it remains unclear about how to solve the problem (11) because we do not know all the weak classifiers. The number of possible weak classifiers could be infinite—the dimension of the optimization variable $\mathbf{w}$ is infinite. So (11) is a semi-infinite quadratic program (SIQP). We show how column generation can be used to solve this problem. To make column generation applicable, we need to derive a specific Lagrange dual of the primal problem.

The Lagrange dual problem. We now derive the Lagrange dual of the quadratic problem (11). Although we are only interested in the variable $\mathbf{w}$, we need to keep the auxiliary variable $\boldsymbol{\rho}$ in order to obtain a meaningful dual problem. The Lagrangian of (11) is $L(\underbrace{\mathbf{w}, \boldsymbol{\rho}}_{\text{primal}}, \underbrace{\mathbf{u}, r}_{\text{dual}}) = \frac{1}{2}\boldsymbol{\rho}^\top \mathbf{Q}\boldsymbol{\rho} - \theta \boldsymbol{e}^\top \boldsymbol{\rho} + \mathbf{u}^\top(\boldsymbol{\rho} - \mathbf{A}\mathbf{w}) - \mathbf{q}^\top \mathbf{w} + r(\mathbf{1}^\top \mathbf{w} - 1)$ with $\mathbf{q} \succcurlyeq \mathbf{0}$. $\sup_{\mathbf{u},r} \inf_{\mathbf{w},\boldsymbol{\rho}} L(\mathbf{w}, \boldsymbol{\rho}, \mathbf{u}, r)$ gives the following Lagrange dual:

$$\max_{\mathbf{u},r} \; -r - \overbrace{\tfrac{1}{2}(\mathbf{u} - \theta \boldsymbol{e})^\top \mathbf{Q}^{-1}(\mathbf{u} - \theta \boldsymbol{e})}^{\text{regularization}}, \quad \text{s.t. } \sum_{i=1}^{m} u_i \mathbf{A}_{i:} \preccurlyeq r\mathbf{1}^\top. \tag{12}$$

In our case, $\mathbf{Q}$ is rank-deficient and its inverse does not exist (for both LDA and LAC). We can simply regularize $\mathbf{Q}$ with $\mathbf{Q} + \delta\mathbf{I}$ with δ a very small constant. One of the KKT optimality conditions between the dual and primal is $\boldsymbol{\rho}^\star = -\mathbf{Q}^{-1}(\mathbf{u}^\star - \theta\boldsymbol{e})$, which can be used to establish the connection between the dual optimum and the primal optimum. This is obtained by the fact that the gradient of L w.r.t. $\boldsymbol{\rho}$ must vanish at the optimum, $\partial L/\partial\rho_i = 0$, $\forall i = 1 \cdots n$.

Problem (12) can be viewed as a regularized LPBoost problem. Compared with the hard-margin LPBoost [5], the only difference is the regularization term in the cost function. The duality gap between the primal (11) and the dual (12) is zero. In other words, the solutions of (11) and (12) coincide. Instead of solving (11) directly, one calculates the most violated constraint in (12) iteratively for the current solution and adds this constraint to the optimization problem. In theory, any column that violates dual feasibility can be added. To speed up the convergence, we add the most violated constraint by solving the following problem:

$$h'(\cdot) = \text{argmax}_{h(\cdot)} \sum_{i=1}^{m} u_i y_i h(\mathbf{x}_i). \tag{13}$$

This is exactly the same as the one that standard AdaBoost and LPBoost use for producing the best weak classifier. That is to say, to find the weak classifier that has minimum weighted training error. We summarize the LACBoost/FisherBoost algorithm in Algorithm 1. By simply changing $\mathbf{Q}_2$, Algorithm 1 can be used to train either LACBoost or FisherBoost. Note that to obtain an actual strong classifier, one may need to include an offset b, *i.e.* the final classifier is $\sum_{j=1}^{n} h_j(\mathbf{x}) - b$ because from the cost function of our algorithm (7), we can see that the cost function itself does not minimize any classification error. It only finds a projection direction in which the data can be maximally separated. A simple line search can find an optimal b. Moreover, when training a cascade, we need to tune this offset anyway as shown in (5).

The convergence of Algorithm 1 is guaranteed by general column generation or cutting-plane algorithms, which is easy to establish. When a new $h'(\cdot)$ that violates dual feasibility is added, the new optimal value of the dual problem (maximization) would decrease. Accordingly, the optimal value of its primal problem decreases too because they have the same optimal value due to zero duality gap. Moreover the primal cost function is convex, therefore in the end it converges to the global minimum.

At each iteration of column generation, in theory, we can solve either the dual (12) or the primal problem (11). However, in practice, it could be much faster to solve the primal problem because (i) Generally, the primal problem has a smaller size, hence faster to solve. The number of variables of (12) is m at each iteration, while the number of variables is the number of iterations for the primal problem. For example, in Viola-Jones' face detection framework, the number of training data $m = 10,000$ and $n_{\max} = 200$. In other words, the primal problem has at most 200 variables in this case; (ii) The dual problem is a standard QP problem. It has no special structure to exploit. As we will show, the primal problem belongs to a special class of problems and can be efficiently

Algorithm 1. Column generation for QP.

Input: Labeled training data $(\mathbf{x}_i, y_i), i = 1 \cdots m$; termination threshold $\varepsilon > 0$; regularization parameter θ; maximum number of iterations $n_{\max}$.

Initialization: $m = 0$; $\mathbf{w} = \mathbf{0}$; and $u_i = \frac{1}{m}$, $i = 1 \cdots m$.

for iteration $= 1 : n_{\max}$ **do**

- Check for the optimality:
  if iteration > 1 and $\sum_{i=1}^{m} u_i y_i h'(\mathbf{x}_i) < r + \varepsilon$,
  then
  break; and the problem is solved;
- Add $h'(\cdot)$ to the restricted master problem, which corresponds to a new constraint in the dual;
- Solve the dual problem (12) (or the primal problem (11)) and update r and u_i $(i = 1 \cdots m)$.
- Increment the number of weak classifiers $n = n + 1$.

Output: The selected features are $h_1, h_2, \ldots, h_n$. The final strong classifier is: $F(\mathbf{x}) = \sum_{j=1}^{n} w_j h_j(\mathbf{x}) - b$. Here the offset b can be learned by a simple search.

solved using entropic/exponentiated gradient descent (EG) [12,13]. A fast QP solver is extremely important for training a object detector because we need to the solve a few thousand QP problems.

We can recover both of the dual variables $\mathbf{u}^\star, r^\star$ easily from the primal variable $\mathbf{w}^\star$:

$$\mathbf{u}^\star = -\mathbf{Q}\boldsymbol{\rho}^\star + \theta \boldsymbol{e}; \tag{14}$$

$$r^\star = \max_{j=1\ldots n} \left\{ \sum_{i=1}^{m} u_i^\star \mathbf{A}_{ij} \right\}. \tag{15}$$

The second equation is obtained by the fact that in the dual problem's constraints, at optimum, there must exist at least one $u_i^\star$ such that the equality holds. That is to say, $r^\star$ is the largest *edge* over all weak classifiers.

We give a brief introduction to the EG algorithm before we proceed. Let us first define the unit simplex $\Delta_n = \{\mathbf{w} \in \mathbb{R}^n : \mathbf{1}^\top \mathbf{w} = 1, \mathbf{w} \succcurlyeq \mathbf{0}\}$. EG efficiently solves the convex optimization problem

$$\min_{\mathbf{w}} \ f(\mathbf{w}), \text{ s.t. } \mathbf{w} \in \Delta_n, \tag{16}$$

under the assumption that the objective function $f(\cdot)$ is a convex Lipschitz continuous function with Lipschitz constant L_f w.r.t. a fixed given norm $\|\cdot\|$. The mathematical definition of L_f is that $|f(\mathbf{w}) - f(\mathbf{z})| \leq L_f \|\mathbf{x} - \mathbf{z}\|$ holds for any $\mathbf{x}, \mathbf{z}$ in the domain of $f(\cdot)$. The EG algorithm is very simple:

1. Initialize with $\mathbf{w}^0 \in$ the interior of Δ_n;
2. Generate the sequence $\{\mathbf{w}^k\}$, $k = 1, 2, \cdots$ with:

$$\mathbf{w}_j^k = \frac{\mathbf{w}_j^{k-1} \exp[-\tau_k f_j'(\mathbf{w}^{k-1})]}{\sum_{j=1}^{n} \mathbf{w}_j^{k-1} \exp[-\tau_k f_j'(\mathbf{w}^{k-1})]}. \tag{17}$$

Here τ_k is the step-size. $f'(\mathbf{w}) = [f_1'(\mathbf{w}), \ldots, f_n'(\mathbf{w})]^\top$ is the gradient of $f(\cdot)$;

3. Stop if some stopping criteria are met.

The learning step-size can be determined by $\tau_k = \frac{\sqrt{2\log n}}{L_f}\frac{1}{\sqrt{k}}$, following [12]. In [13], the authors have used a simpler strategy to set the learning rate.

EG is a very useful tool for solving large-scale convex minimization problems over the unit simplex. Compared with standard QP solvers like Mosek [14], EG is much faster. EG makes it possible to train a detector using almost the same amount of time as using standard AdaBoost as the majority of time is spent on weak classifier training and bootstrapping.

In the case that $m_1 \gg 1$,

$$\mathbf{Q}_1 = \frac{1}{m}\begin{bmatrix} 1 & -\frac{1}{m_1-1} & \cdots & -\frac{1}{m_1-1} \\ -\frac{1}{m_1-1} & 1 & \cdots & -\frac{1}{m_1-1} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{m_1-1} & -\frac{1}{m_1-1} & \cdots & 1 \end{bmatrix} \approx \frac{1}{m}\mathbf{I}.$$

Similarly, for LDA, $\mathbf{Q}_2 \approx \frac{1}{m}\mathbf{I}$ when $m_2 \gg 1$. Hence,

$$\mathbf{Q} \approx \begin{cases} \frac{1}{m}\mathbf{I}; & \text{for Fisher LDA,} \\ \frac{1}{m}\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, & \text{for LAC.} \end{cases} \tag{18}$$

Therefore, the problems involved can be simplified when $m_1 \gg 1$ and $m_2 \gg 1$ hold. The primal problem (11) equals

$$\min_{\mathbf{w},\boldsymbol{\rho}} \ \tfrac{1}{2}\mathbf{w}^\top(\mathbf{A}^\top\mathbf{Q}\mathbf{A})\mathbf{w} - (\theta\boldsymbol{e}^\top\mathbf{A})\mathbf{w}, \ \text{ s.t. } \mathbf{w} \in \Delta_n. \tag{19}$$

We can efficiently solve (19) using the EG method. In EG there is an important parameter L_f, which is used to determine the step-size. L_f can be determined by the ℓ_∞-norm of $|f'(\mathbf{w})|$. In our case $f'(\mathbf{w})$ is a linear function, which is trivial to compute. The convergence of EG is guaranteed; see [12] for details.

In summary, when using EG to solve the primal problem, Line 5 of Algorithm 1 is:

– *Solve the primal problem* (19) *using EG, and update the dual variables* $\mathbf{u}$ *with* (14)*, and* r *with* (15)*.*

4 Applications to Face Detection

First, let us show a simple example on a synthetic dataset (more negative data than positive data) to illustrate the difference between FisherBoost and AdaBoost. Fig. 1 demonstrates the subtle difference of the classification boundaries obtained by AdaBoost and FisherBoost. We can see that FisherBoost seems to focus more on correctly classifying positive data points. This might be due to the fact that AdaBoost only optimizes the overall classification accuracy. This finding is consistent with the result in [15].

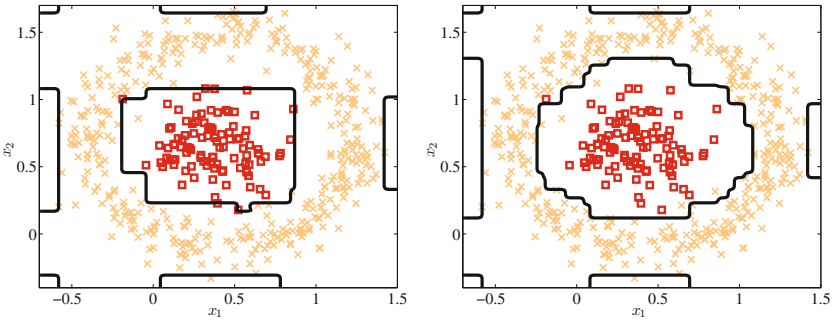


Fig. 1. Decision boundaries of AdaBoost (left) and FisherBoost (right) on 2D artificial data (positive data represented by □'s and negative data by ×'s). Weak classifiers are decision stumps. In this case, FisherBoost tends to correctly classify more positive data in this case.

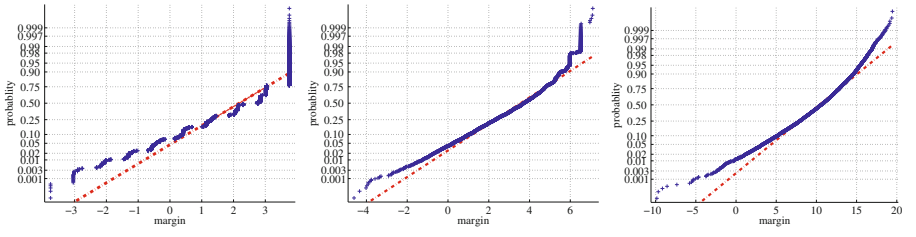


Fig. 2. Normality test (normal probability plot) for the face data's margin distribution of nodes 1, 2, 3. The 3 nodes contains 7, 22, 52 weak classifiers respectively. Curves close to a straight line mean close to a Gaussian.

Face detection. In this section, we compare our algorithm with other state-of-art face detectors. We first show some results about the validity of LAC (or Fisher LDA) post-processing for improving node learning in object detection. Fig. 2 illustrates the normal probability plot of margins of positive training data, for the first three nodes in the multi-exit with LAC cascade. Clearly, the larger number of weak classifiers being used, the more closely the margin follows Gaussian distribution. In other words, LAC may achieve a better performance if a larger number of weak classifiers are used. The performance could be poor with too fewer weak classifiers. The same statement applies to Fisher LDA, and LACBoost, FisherBoost, too. Therefore, we do not apply LAC/LDA in the first eight nodes because the margin distribution could be far from a Gaussian distribution. Because the late nodes of a multi-exit cascade contain more weak classifiers, we conjecture that the multi-exit cascade might meet the Gaussianity requirement better. We have compared multi-exit cascades with LDA/LAC post-processing against standard cascades with LDA/LAC post-processing in [4] and slightly improved performances were obtained.

Six methods are evaluated with the multi-exit cascade framework [8], which are AdaBoost with LAC post-processing, or LDA post-processing, AsymBoost with LAC or LDA post-processing [4], and our FisherBoost, LACBoost. We have also implemented Viola-Jones' face detector as the baseline [2]. As in [2], five basic types of Haar-like features are calculated, which makes up of a $162,336$ dimensional over-complete feature set on an image of 24×24 pixels. To speed up the weak classifier training, as in [4], we uniformly sample 10% of features for training weak classifiers (decision stumps). The training data are $9,832$ mirrored 24×24 face images ($5,000$ for training and $4,832$ for validation) and $7,323$ large background images, which are the same as in [4].

Multi-exit cascades with 22 exits and $2,923$ weak classifiers are trained with various methods. For fair comparisons, we have used the same cascade structure and same number of weak classifiers for all the compared learning methods. The indexes of exits are pre-set to simplify the training procedure. For our FisherBoost and LACBoost, we have an important parameter θ, which is chosen from $\{\frac{1}{10}, \frac{1}{12}, \frac{1}{15}, \frac{1}{20}, \frac{1}{25}, \frac{1}{30}, \frac{1}{40}, \frac{1}{50}\}$. We have not carefully tuned this parameter using cross-validation. Instead, we train a 10-node cascade for each candidate θ, and choose the one with the best *training* accuracy.[2] At each exit, negative examples misclassified by current cascade are discarded, and new negative examples are bootstrapped from the background images pool. Totally, billions of negative examples are extracted from the pool. The positive training data and validation data keep unchanged during the training process.

Our experiments are performed on a workstation with 8 Intel Xeon E5520 CPUs and 32GB RAM. It takes about 3 hours to train the multi-exit cascade with AdaBoost or AsymBoost. For FisherBoost and LACBoost, it takes less than 4 hours to train a complete multi-exit cascade.[3] In other words, our EG algorithm takes less than 1 hour for solving the primal QP problem (we need to solve a QP at each iteration). A rough estimation of the computational complexity is as follows. Suppose that the number of training examples is m, number of weak classifiers is n, At each iteration of the cascade training, the complexity for solving the primal QP using EG is $O(mn + kn^2)$ with k the iterations needed for EQ's convergence. The complexity for training the weak classifier is $O(md)$ with d the number of all Haar-feature patterns. In our experiment, $m = 10,000$, $n \approx 2900$, $d = 160,000$, $k < 500$. So the majority of the training computation is on the weak classifier training.

We have also experimentally observed the speedup of EG against standard QP solvers. We solve the primal QP defined by (19) using EG and Mosek [14]. The QP's size is $1,000$ variables. With the same accuracy tolerance (Mosek's primal-dual gap is set to 10^{-7} and EG's convergence tolerance is also set to 10^{-7}), Mosek takes 1.22 seconds and EG is 0.0541 seconds. So EG is about 20 times faster. Moreover, at iteration $n+1$ of training the cascade, EG can take advantage of the

[2] To train a complete 22-node cascade and choose the best θ on cross-validation data may give better detection rates.

[3] Our implementation is in C++ and only the weak classifier training part is parallelized using OpenMP.

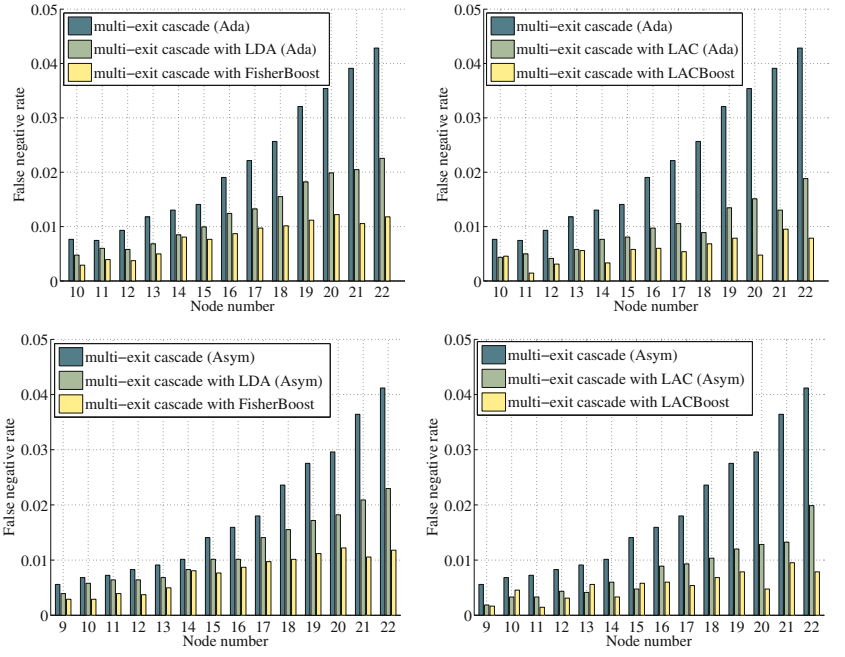


Fig. 3. Node performances on the validation data. "Ada" means that features are selected using AdaBoost; "Asym" means that features are selected using AsymBoost.

last iteration's solution by starting EG from a small perturbation of the previous solution. Such a warm-start gains a 5 to 10× speedup in our experiment, while there is no off-the-shelf warm-start QP solvers available yet.

We evaluate the detection performance on the MIT+CMU frontal face test set. Two performance metrics are used here: each node and the entire cascade. The node metric is how well the classifiers meet the node learning objective. The node metric provides useful information about the capability of each method to achieve the node learning goal. The cascade metric uses the receiver operating characteristic (ROC) to compare the entire cascade's peformance. Multiple issues have impacts on the cascade's performance: classifiers, the cascade structure, bootstrapping *etc.*

We show the node comparison results in Fig. 3. The node performances between FisherBoost and LACBoost are very similar. From Fig. 3, as reported in [4], LDA or LAC post-processing can considerably reduce the false negative rates. As expected, our proposed FisherBoost and LACBoost can further reduce the false negative rates significantly. This verifies the advantage of selecting features with the node learning goal being considered.

From the ROC curves in Fig. 4, we can see that FisherBoost and LACBoost outperform all the other methods. In contrast to the results of the detection rate for each node, LACBoost is slightly worse than FisherBoost in some cases.

That might be due to that many factors have impacts on the final result of detection. LAC makes the assumption of Gaussianity and symmetry data distributions, which may not hold well in the early nodes. This could explain why LACBoost does not always perform the best. Wu *et al.* have observed the same phenomenon that LAC post-processing does not outperform LDA post-processing in a few cases. However, we believe that for harder detection tasks, the benefits of LACBoost would be more impressive.

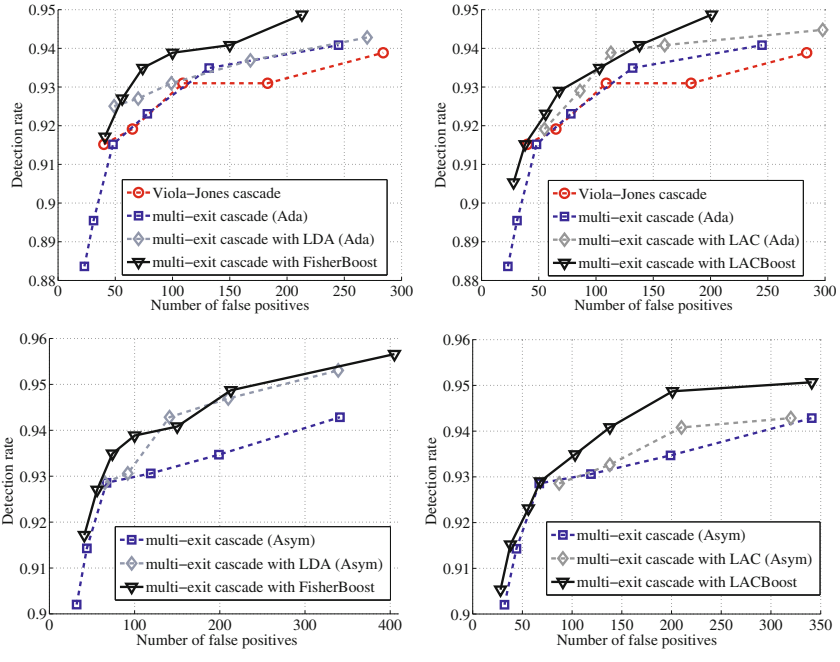


Fig. 4. Cascade performances using ROC curves (number of false positives versus detection rate) on the MIT+CMU test data. "Ada" means that features are selected using AdaBoost. Viola-Jones cascade is the method in [2]. "Asym" means that features are selected using AsymBoost.

The error reduction results of FisherBoost and LACBoost in Fig. 4 are not as great as those in Fig. 3. This might be explained by the fact that the cascade and negative data bootstrapping remove of the error reducing effects, to some extent. We have also compared our methods with the boosted greedy sparse LDA (BGSLDA) in [15], which is considered one of the state-of-the-art. We provide the ROC curves in the supplementary package. Both of our methods outperform BGSLDA with AdaBoost/AsymBoost by about 2% in the detection rate. Note that BGSLDA uses the standard cascade. So besides the benefits of our FisherBoost/LACBoost, the multi-exit cascade also brings effects.

5 Conclusion

By explicitly taking into account the node learning goal in cascade classifiers, we have designed new boosting algorithms for more effective object detection. Experiments validate the superiority of our FisherBoost and LACBoost. We have also proposed to use entropic gradient to efficiently implement FisherBoost and LACBoost. The proposed algorithms are easy to implement and can be applied other asymmetric classification tasks in computer vision. We are also trying to design new asymmetric boosting algorithms by looking at those asymmetric kernel classification methods.

References

1. Wu, J., Mullin, M.D., Rehg, J.M.: Linear asymmetric classifier for cascade detectors. In: Proc. Int. Conf. Mach. Learn., Bonn, Germany, pp. 988–995 (2005)
2. Viola, P., Jones, M.J.: Robust real-time face detection. Int. J. Comp. Vis. 57(2), 137–154 (2004)
3. Viola, P., Jones, M.: Fast and robust classification using asymmetric AdaBoost and a detector cascade. In: Proc. Adv. Neural Inf. Process. Syst., pp. 1311–1318. MIT Press, Cambridge (2002)
4. Wu, J., Brubaker, S.C., Mullin, M.D., Rehg, J.M.: Fast asymmetric learning for cascade face detection. IEEE Trans. Pattern Anal. Mach. Intell. 30(3), 369–382 (2008)
5. Demiriz, A., Bennett, K., Shawe-Taylor, J.: Linear programming boosting via column generation. Mach. Learn. 46(1-3), 225–254 (2002)
6. Bourdev, L., Brandt, J.: Robust object detection via soft cascade. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., San Diego, CA, US, pp. 236–243 (2005)
7. Xiao, R., Zhu, H., Sun, H., Tang, X.: Dynamic cascades for face detection. In: Proc. IEEE Int. Conf. Comp. Vis., Rio de Janeiro, Brazil (2007)
8. Pham, M.T., Hoang, V.D.D., Cham, T.J.: Detection with multi-exit asymmetric boosting. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., Anchorage, Alaska (2008)
9. Li, S.Z., Zhang, Z.: FloatBoost learning and statistical face detection. IEEE Trans. Pattern Anal. Mach. Intell. 26(9), 1112–1123 (2004)
10. Liu, C., Shum, H.Y.: Kullback-Leibler boosting. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., Madison, Wisconsin, vol. 1, pp. 587–594 (June 2003)
11. Rätsch, G., Mika, S., Schölkopf, B., Müller, K.R.: Constructing boosting algorithms from SVMs: An application to one-class classification. IEEE Trans. Pattern Anal. Mach. Intell. 24(9), 1184–1199 (2002)
12. Beck, A., Teboulle, M.: Mirror descent and nonlinear projected subgradient methods for convex optimization. Oper. Res. Lett. 31(3), 167–175 (2003)
13. Collins, M., Globerson, A., Koo, T., Carreras, X., Bartlett, P.L.: Exponentiated gradient algorithms for conditional random fields and max-margin Markov networks. J. Mach. Learn. Res., 1775–1822 (2008)
14. MOSEK ApS: The MOSEK optimization toolbox for matlab manual, version 5.0, revision 93 (2008), http://www.mosek.com/
15. Paisitkriangkrai, S., Shen, C., Zhang, J.: Efficiently training a better visual detector with sparse Eigenvectors. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., Miami, Florida, US (June 2009)

A Shrinkage Learning Approach for Single Image Super-Resolution with Overcomplete Representations

Amir Adler[1], Yacov Hel-Or[2], and Michael Elad[1]

[1] Computer Science Department, The Technion, Haifa, Israel
[2] Efi Arazi School of Computer Science,
The Interdisciplinary Center, Herzelia, Israel

Abstract. We present a novel approach for online shrinkage functions learning in single image super-resolution. The proposed approach leverages the classical *Wavelet Shrinkage* denoising technique where a set of scalar shrinkage functions is applied to the wavelet coefficients of a noisy image. In the proposed approach, a unique set of learned shrinkage functions is applied to the overcomplete representation coefficients of the interpolated input image. The super-resolution image is reconstructed from the post-shrinkage coefficients. During the learning stage, the low-resolution input image is treated as a reference high-resolution image and a super-resolution reconstruction process is applied to a scaled-down version of it. The shapes of all shrinkage functions are jointly learned by solving a Least Squares optimization problem that minimizes the sum of squared errors between the reference image and its super-resolution approximation. Computer simulations demonstrate superior performance compared to state-of-the-art results.

1 Introduction

Single Image Super-Resolution (SISR) is the process of reconstructing a high-resolution image from an observed low-resolution image. Typical applications include zoom-in of still images in digital cameras, scaling-up an image before printing and conversion from low-definition to high-definition video. SISR is an inverse problem, associated with the following linear degradation model

$$\mathbf{y} = DH\mathbf{x}, \tag{1}$$

where $\mathbf{y} \in \mathbb{R}^n$ is the observed low-resolution input image (column-stacked), $\mathbf{x} \in \mathbb{R}^{nL}$ is the unknown high-resolution image, $H \in \mathbb{R}^{nL \times nL}$ is a blurring filter (block-circulant) convolution matrix and $D \in \mathbb{R}^{n \times nL}$ is a down-sample operator matrix, decimating the image by a factor of $\sqrt{L}$ along the horizontal and vertical dimensions.

A solution to the SISR problem is an approximation $\hat{\mathbf{x}}$ to the unknown high-resolution image $\mathbf{x}$. Since the linear system (1) is underdetermined, there are infinitely many solutions $\hat{\mathbf{x}}$ that can "explain" the observed image $\mathbf{y}$. For this

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 622–635, 2010.

reason, there are various approaches addressing the SISR problem. The simplest techniques are the bi-linear and bi-cubic interpolators. These interpolators utilize a polynomial approximation model to compute each missing pixel from a small local neighborhood of it, often generating blurry results and stair-case shaped edges.

State-of-the-art SISR reconstruction is based on a sparse-representation approach [1], [2] where a set of high-resolution and low-resolution dictionaries are learned from example images. In this approach, a sparse coding process is applied to small overlapping patches, extracted in a raster-scan order from the observed image. The sparse representation coefficients (i.e the outcome of the sparse coding process) of each low-resolution patch are assumed to faithfully represent each corresponding (unknown) high-resolution patch by replacing the low-resolution dictionary with its high-resolution counterpart. The super-resolution image is reconstructed by fusion of all of the overlapping high-resolution patches. A similar approach was proposed in [3], where a reduced redundancy dictionary was employed to accelerate the SISR process. The sparse-representation approach evolved from an example-based approach [4], where a dictionary of $100,000$ pairs of low-resolution and high-resolution images patches was utilized in conjunction with a markov-network model to search-and-match the corresponding high-resolution patches. A combination of the example-based approach with multi-frame super-resolution was proposed in [5], where patch repetitions within an image were exploited in a multi-scale approach. Additional example-based approaches such as learning the prior parameters, learning the posterior and building example-based regularization expression are reviewed in [6]. A shrinkage-based approach was introduced in [7] where a hard-thresholding function was iteratively applied to DCT transform coefficients. This approach was later augmented in [8], where the Contourlet transform was chosen as the overcomplete transform.

We propose to extend the shrinkage based approach and employ online-learned shrinkage functions with an overcomplete representation. The proposed approach leverages the *discriminative* learning technique suggested in [9] for wavelet denoising. In the discriminative approach, the shapes of all shrinkage functions are learned offline from example images (rather then learning the parameters of a probability distribution model of the transform coefficients). In the proposed approach, we apply the discriminative approach to the SISR problem and exploit the scale-invariant property of natural images [10] to learn the shrinkage functions directly from the input image.

Contributions. The contributions presented in this paper are two-fold: 1) Introduction of the learned shrinkage approach [9] to solve the SISR problem, in contrast to the hard-thresholding approach previously introduced for SISR in [8], [7]. 2) Introduction of the online learning approach, where the shrinkage functions are learned directly from the observed input image - in contrast to the offline example-based approach as suggested in [9]. The advantage of the online approach is that online-learned shrinkage functions capture the statistical properties of the observed image (to be scaled-up), rather than the statistical

properties of other images. Performance evaluation of the proposed approach demonstrate superior performance compared to the sparse representation[1] state-of-the-art approach [1], [2].

This paper is organized as follows: Section 2 presents shrinkage-based restoration theory for the unitary and over-complete cases. Section 3 describes the Slice Transform (SLT) which is a piece-wise linear model utilized for the representation and learning of the shrinkage functions. Section 4 presents the proposed SISR algorithm concept along with a detailed explanation of the shrinkage-learning stage and the super-resolution reconstruction stage. Section 5 overviews performance evaluation along with a comparison versus the state-of-the-art approach.

2 Shrinkage-Based Image Restoration

This section provides an overview of shrinkage-based image restoration in the unitary and overcomplete cases. The discussion evolves from an image denoising problem and the connection to the SISR problem is established in the last subsection. Consider the following image degradation model,

$$\mathbf{v} = \mathbf{u} + \mathbf{m}, \tag{2}$$

where $\mathbf{v} \in \mathbb{R}^l$ is an observed noisy image, $\mathbf{u} \in \mathbb{R}^l$ is the unknown clean image and $\mathbf{m} \in \mathbb{R}^l$ is white Gaussian noise. In the shrinkage-based approach, the restored image is given by the following algorithm

$$\hat{\mathbf{u}} = W^{\dagger} \overrightarrow{\Psi} \left(W\mathbf{v} \right), \tag{3}$$

where W is a unitary or overcomplete transform, $\overrightarrow{\Psi} = [\Psi_1, \Psi_2, \dots]$ is a set of scalar shrinkage functions and $W^{\dagger}$ is the reverse transform. The utilization of scalar shrinkage functions is derived in the following subsections.

2.1 The Unitary Case

The shrinkage-based reconstruction (3) can be shown to solve a MAP estimation problem under the assumptions of a unitary transform, independent transform coefficients and white Gaussian noise. The discussion is focused on the unitary wavelet transform, since it provides a sparse representation of natural images [11] and its coefficients are assumed independent. These properties of the unitary wavelet transform play a fundamental role in the formulation of sparsity-promoting image priors [12] that can be decoupled into a product (or a sum in the log domain) of scalar probability distributions. The MAP estimator $\hat{\mathbf{u}}\left(\mathbf{v}\right)$ is given by maximizing the a-posteriori probability:

$$\hat{\mathbf{u}}\left(\mathbf{v}\right) = \arg\max_{\mathbf{u}} P\left(\mathbf{u} \mid \mathbf{v}\right). \tag{4}$$

[1] In this paper we refer to the work in [1], [2] as "sparse representation" based, although the shrinkage based approach also emerges from sparse representation modeling.

This maximization can be cast also in the transform domain, as follows:

$$\hat{\mathbf{u}}_W(\mathbf{v}_W) = \arg\max_{\mathbf{u}_W} P(\mathbf{u}_W \mid \mathbf{v}_W), \tag{5}$$

where $\mathbf{u}_W = W\mathbf{u}$, $\mathbf{v}_W = W\mathbf{v}$ and W is a unitary wavelet transform. By utilizing *Bayes* rule and the monotonicity of the log function, the wavelet domain MAP estimator can be reformulated as

$$\hat{\mathbf{u}}_W(\mathbf{v}_W) = \arg\min_{\mathbf{u}_W}\{-\log P(\mathbf{v}_W \mid \mathbf{u}_W) - \log P(\mathbf{u}_W)\}. \tag{6}$$

The term $\log P(\mathbf{v}_W \mid \mathbf{u}_W)$ is the *log likelihood* and the term $\log P(\mathbf{u}_W)$ is the *prior*. For the white Gaussian noise case, the log likelihood term is given by

$$-\log P(\mathbf{v}_W \mid \mathbf{u}_W) = \lambda\|\mathbf{u}_W - \mathbf{v}_W\|^2 = \lambda\sum_i \|u_W^i - v_W^i\|^2, \tag{7}$$

where u_W^i and v_W^i are the i-th elements of $\mathbf{u}_W$ and $\mathbf{v}_W$, respectively and λ is a constant inversely proportional to the noise variance. Note, that by utilizing the l^2-norm preserving property of unitary transforms, equation (7) can be rewritten as

$$-\log P(\mathbf{v}_W \mid \mathbf{u}_W) = \lambda\|W(\mathbf{u}-\mathbf{v})\|^2 = \lambda\|\mathbf{u}-\mathbf{v}\|^2 = -\log P(\mathbf{v} \mid \mathbf{u}). \tag{8}$$

Thus, the spatial domain MAP estimator (4) and its unitary transform domain counterpart (5) are equivalent, as long as the *prior* term is a function of $W\mathbf{u}$ [12]. By utilizing the independence assumption of the unitary wavelet coefficients, the *prior* term is reformulated as

$$\log P(\mathbf{u}_W) = \log\prod_i P_i\left(u_W^i\right) = \sum_i \log P_i\left(u_W^i\right). \tag{9}$$

The unitary wavelet domain MAP estimator (6) can be rewritten using the results of equations (7) and (9), leading to a decoupling of the l-dimensional minimization problem to a set of l scalar minimization problems

$$\hat{u}_W^i\left(v_W^i\right) = \arg\min_{u_W^i}\{\lambda\|u_W^i - v_W^i\|^2 - \log P_i\left(u_W^i\right)\} \quad \forall i. \tag{10}$$

The optimization in equation (10) is solved by applying a scalar lookup table function Ψ_W^i, termed *shrinkage* function, to the wavelet coefficients: $\hat{u}_W^i\left(v_W^i\right) = \Psi_W^i\left(v_W^i\right)$. The shrinkage function depends solely on the noise variance and the prior term $P_i\left(u_W^i\right)$. The pioneering studies of Donoho and Johnstone [13], [14] suggested using *hard-thresholding* and *soft-thresholding* shrinkage functions. Furthermore, for a K subband wavelet transform, only K distinct shrinkage functions are required to solve the MAP estimation problem. To clarify this property we follow the notation in [9] and utilize a permutation matrix P to reorder the rows of the wavelet transform W. The reordering is performed such that wavelet

transform rows corresponding to a specific subband are co-located in a distinct block

$$B = PW = \begin{bmatrix} B_1 \\ \vdots \\ B_K \end{bmatrix} \quad \text{and} \quad \mathbf{v_B} = B\mathbf{v} = \begin{bmatrix} \mathbf{v_{B_1}} \\ \vdots \\ \mathbf{v_{B_K}} \end{bmatrix}. \tag{11}$$

The set of K shrinkage functions are denoted by $\overrightarrow{\Psi_B} = [\Psi_{B_1}, \Psi_{B_2}, \ldots, \Psi_{B_K}]$ and the restored image (3) is given by

$$\hat{\mathbf{u}} = B^T \overrightarrow{\Psi_B}\{\mathbf{v_B}\} = \sum_{k=1}^{K} B_k^T \Psi_{B_k}\{\mathbf{v_{B_k}}\}, \tag{12}$$

where B^T is the reverse transform due to the unitary case assumption.

2.2 The Overcomplete Case

The shrinkage restoration approach in the unitary case provides good results, however, visual artifacts sometimes appear in the restored image. By utilizing an overcomplete transform, significant improvements can be achieved. This was originally discovered by Coifman and Donoho [15] where an undecimated wavelet transform provided superior shrinkage denoising results compared to the unitary case. This improvement was later demonstrated in various overcomplete transforms such a Curvelets [16], Contourlets [17], undecimated windowed DCT [9] and others. By applying equation (11) to the overcomplete case, the noisy image transform is given by $\mathbf{v_B} = B\mathbf{v}$. The overcomplete transform B is an $M \times l$ matrix where $M > l$. By modifying $\mathbf{v_B}$ using a vector of shrinkage functions $\overrightarrow{\Psi_B}\{\mathbf{v_B}\}$ it is desired that all the post-shrinkage overcomplete components be equal to the overcomplete transform components of the original (unknown) image

$$B\mathbf{u} = \overrightarrow{\Psi_B}\{\mathbf{v_B}\}. \tag{13}$$

The estimated image is reconstructed using the pseudo-inverse

$$\hat{\mathbf{u}} = \left(B^T B\right)^{-1} B^T \overrightarrow{\Psi_B}\{\mathbf{v_B}\} = \left(B^T B\right)^{-1} \sum_{k=1}^{K} B_k^T \Psi_{B_k}\{\mathbf{v_{B_k}}\}. \tag{14}$$

A key difference between the unitary and overcomplete cases is statistical dependence of the transform coefficients: the scalar shrinkage approach emerged from the independence assumption of the unitary wavelet coefficients, however, this assumption no longer holds in the overcomplete case. Traditionally, the unitary case shrinkage functions were applied also to the overcomplete case, however, the interband dependencies of the overcomplete transform coefficients should be taken into account. The most accurate approach to handle this issue is to design a set of multi-dimensional shrinkage functions, however, such approach is highly complex. The approach suggested in [9] for image denoising is to learn a set of *scalar* shrinkage functions that would take into account *interband* as well

as *intraband* dependencies. In this approach, the shrinkage functions are learned offline from an example set of pairs of clean and noisy images. In this paper, we leverage the shrinkage learning technique to the SISR problem and propose to learn the shrinkage functions online - from the observed image - in a way that would capture the statistical properties of (only) the image to be scaled-up.

2.3 From Image Denoising to Super-Resolution

The shrinkage-based restoration framework was originally developed for image denoising. However, it has been successfully utilized for more complex inverse problems then (2), by designing the shrinkage operation to minimize all structured noise components inherent to the specific problem. For example, inpainting by hard-thresholding [18], SISR by hard-thresholding [7], [8] and JPEG deblocking [9]. We Assume a general image degradation model:

$$\mathbf{v} = \Omega\{\mathbf{u}\} = \mathbf{u} + \mathbf{e}, \tag{15}$$

where $\Omega\{\cdot\}$ is a degradation operator (not necessarily linear) and $\mathbf{e}$ is an error image with unknown statistical properties. We propose to recover the unknown image $\mathbf{u}$ by utilizing the restoration algorithm (14), with a set of shrinkage functions that were designed to maximize the restored image quality, given the degradation model (15). For the SISR problem, we utilize the following degradation operator

$$\Omega\{\mathbf{u}\} = \Upsilon_{\uparrow}\left(DH\mathbf{u}\right), \tag{16}$$

where $\Upsilon_{\uparrow}\left(\cdot\right)$ is a simple interpolator (implemented either by a bi-linear or bi-cubic interpolator). Note, that this degradation operator simply amounts to an interpolation of the observed image $\mathbf{y}$ in the SISR model (1) and the dimensions of the degraded image $\mathbf{v} = \Omega\{\mathbf{u}\} = \Upsilon_{\uparrow}\left(DH\mathbf{u}\right)$ are identical to $\mathbf{u}$. Therefore, the proposed restoration scheme for the SISR problem is as follows

$$\hat{\mathbf{u}} = \left(B^T B\right)^{-1} B^T \overrightarrow{\Psi_B}\{\mathbf{v_B}\} = \left(B^T B\right)^{-1} \sum_{k=1}^{K} B_k^T \Psi_{B_k}\{\mathbf{v_{B_k}}\}. \tag{17}$$

In the proposed approach, the shapes of all shrinkage functions $\overrightarrow{\Psi_B}$ are trained for the SISR problem. The training is performed online (i.e. directly) from the observed image $DH\mathbf{u}$, exploiting the scale-invariant property of natural images [10]. The learning procedure relies on a piece-wise linear model of the shrinkage functions as explained in the following section. The learning process is explained in section 4.

3 The Slice Transform

The Slice Transform (SLT) [9] enables the approximation of a shrinkage function in a linear manner

$$\Psi_{B_k}\{\mathbf{v}_{B_k}\} \approx S_{\mathbf{q}_k}\left(\mathbf{v}_{B_k}\right)\mathbf{p}_k. \tag{18}$$

Note, that while the shrinkage function is a scalar function, the representation (18) incorporates the element-wise shrinkage operation for the entire subband B_k. The i-th row of the sparse matrix $S_{\mathbf{q}_k}(\mathbf{v}_{B_k})$ is determined uniquely by the i-th element of the vector $\mathbf{v}_{\mathbf{B_k}}$ and the predefined vector $\mathbf{q}_k$. The vector $\mathbf{p}_k$ is the design parameter that controls the input-output mapping relation of the k-th shrinkage function. In the following we explain the concept behind the representation (18) and begin, for simlicity, with the scalar case.

Assume $x \in [a,b)$ is a real value and the half open interval $[a,b)$ is divided into M slots. The boundaries of the slots are contained in the vector $\mathbf{q} = [q_0, q_1, \ldots, q_M]^T$ such that $q_0 = a < q_1 < q_2 \cdots < q_M = b$. The value x is located in a single slot $\pi(x) \in \{1, \ldots, M\}$ and associated with a residue $r(x)$, where $\pi(x) = j$ if $x \in [q_{j-1}, q_j)$ and

$$r(x) = \frac{x - q_{\pi(x)-1}}{q_{\pi(x)} - q_{\pi(x)-1}}$$

Note that $r(x) \in [0,1)$, where $r(x) = 0$ if $x = q_{\pi(x)-1}$ and $r(x) \to 1$ if $x \to q_{\pi(x)}$. The value x can be expressed as a follows

$$x = S_{\mathbf{q}}(x)\,\mathbf{q} = r(x)\,q_{\pi(x)} + (1 - r(x))\,q_{\pi(x)-1}, \tag{19}$$

where the row vector $S_{\mathbf{q}}(x) \in \mathbb{R}^{(M+1)}$ is defined as follows:

$$S_{\mathbf{q}}(x) = [0, \ldots, 0, 1 - r(x), r(x), 0, \ldots, 0]$$

and where the values $1 - r(x)$ and $r(x)$ are located in the $(\pi(x) - 1)^{th}$ and $(\pi(x))^{th}$ entries, respectively. Extending equation (19) to the multi-dimensional case, we assume that $\mathbf{x} \in \mathbb{R}^N$ and that each element satisfies $x^i \in [a,b)$. The SLT of $\mathbf{x}$ is given by

$$\mathbf{x} = S_{\mathbf{q}}(\mathbf{x})\,\mathbf{q}, \tag{20}$$

where the matrix $S_{\mathbf{q}}(\mathbf{x}) \in \mathbb{R}^{N \times (M+1)}$ is given by

$$[S_{\mathbf{q}}(\mathbf{x})]_{i,j} = \begin{cases} r(x^i) & \text{if } \pi(x^i) = j \\ 1 - r(x^i) & \text{if } \pi(x^i) = j+1 \\ 0 & \text{otherwise.} \end{cases} \tag{21}$$

Each row of the matrix $S_{\mathbf{q}}(\mathbf{x})$ is associated with a single element of the vector $\mathbf{x}$ and the representation (20) is composed of linear splines basis functions. According to [9], substituting the boundary vector $\mathbf{q}$ with a different vector $\mathbf{p}$ performs a piece-wise linear mapping of the values in $\mathbf{x}$

$$\mathbf{M}_{\mathbf{q},\mathbf{p}}(\mathbf{x}) = S_{\mathbf{q}}(\mathbf{x})\,\mathbf{p}, \tag{22}$$

where $\mathbf{M}_{\mathbf{q},\mathbf{p}}(\mathbf{x})$ performs linear mapping of the values $\{x^i \in [q_{j-1}, q_j)\}$ to the interval $[p_{j-1}, p_j)$, as depicted in Fig. 1. The substitution property (22) is the key principal behind the linear representation of the shrinkage functions (18).

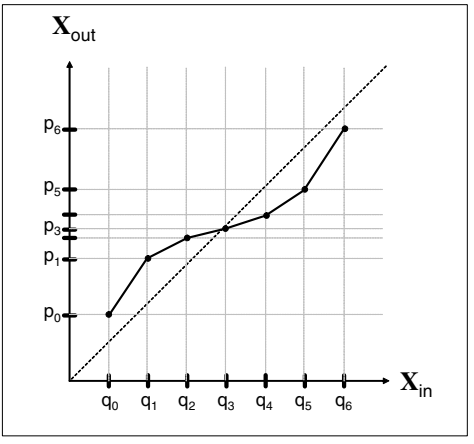


Fig. 1. Piece-wise linear mapping with the Slice Transform

4 The Super-Resolution Algorithm

The proposed Super-Resolution algorithm includes two stages: during the first stage a pair of example images are utilized in an online discriminative learning process of the shrinkage functions. In the second stage, the learned shrinkage functions are applied during the super-resolution reconstruction.

4.1 Stage I: Learning the Shrinkage Functions

The shrinkage functions learning algorithm is inspired by an *oracle* based approach. Consider the SISR degradation model (1) and an oracle estimator of the shrinkage functions that has access to the input image $\mathbf{y}$ and to the unknown high-resolution image $\mathbf{x}$. The oracle learning strategy is based on constructing a super-resolution approximation $\hat{\mathbf{x}}$ from the interpolated low-resolution image $\mathbf{y}_{\uparrow} = \Upsilon_{\uparrow}(\mathbf{y})$ by employing the scheme in (17) such that the unknown shrinkage functions are represented by the SLT approximation (18)

$$\hat{\mathbf{x}}(\mathbf{y}_{\uparrow}, \mathbf{p}) = \left(B^T B\right)^{-1} \sum_{k=1}^{K} B_k^T S_{\mathbf{q}_k}(\mathbf{y}_{\uparrow B_k}) \mathbf{p}_k = L(\mathbf{y}_{\uparrow}) \mathbf{p}, \tag{23}$$

where $\mathbf{p} = \left[\mathbf{p_1^T}, \mathbf{p_2^T}, \ldots, \mathbf{p_K^T}\right]^{\mathbf{T}}$ and

$$\mathbf{L}(\mathbf{y}_{\uparrow}) = \left(\mathbf{B^T B}\right)^{-1} [\mathbf{H_1}, \mathbf{H_2}, \ldots, \mathbf{H_K}]$$

where $\mathbf{H_i} = \mathbf{B_i^T S_{q_i}}(\mathbf{y}_{\uparrow \mathbf{B_i}})$. The oracle learns the unknown shrinkage functions by solving the following Least Squares (LS) optimization problem

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \|\mathbf{x} - \hat{\mathbf{x}}(\mathbf{y}_{\uparrow}, \mathbf{p})\|_{\mathbf{2}}^{\mathbf{2}}. \tag{24}$$

This ideal strategy captures all interband and intraband statistical dependencies of the reconstructed image such that the spatial domain mean squared error (MSE) between the reconstructed and true images is minimized. In practice, only the observed low-resolution image is available and a question arises - can we learn the shrinkage functions in a similar fashion to the oracle with only $\mathbf{y}$ at hand? Here we exploit the scale-invariant property of natural images [10] and we approximate it by the following approach: the oracle training pair $\{\mathbf{x}, \mathbf{y}_{\uparrow}\}$ is replaced with the pair $\{\mathbf{y}, \mathbf{g}\}$ such that the reference image is now the observed low-resolution image and its degraded counterpart is given by

$$\mathbf{g} = \Upsilon_{\uparrow}\left(\widetilde{D}\widetilde{H}\mathbf{y}\right) \in \mathbb{R}^{n},$$

where $\widetilde{\mathbf{H}} \in \mathbb{R}^{\mathbf{n}\times\mathbf{n}}$ is a blurring filter (block-circulant) convolution matrix and $\widetilde{\mathbf{D}} \in \mathbb{R}^{\frac{\mathbf{n}}{\mathbf{L}}\times\mathbf{n}}$ is a down-sampling operator matrix, by a factor of $\sqrt{L}$ along the horizontal and vertical dimensions. Thus, the super-resolution reconstruction is applied to a scaled-down version of the low-resolution observed image

$$\hat{\mathbf{y}}\left(\mathbf{g}, \mathbf{p}\right) = \left(B^{T}B\right)^{-1} \sum_{k=1}^{K} B_{k}^{T} S_{\mathbf{q}_k}\left(\mathbf{g}_{B\,k}\right) \mathbf{p}_k = L\left(\mathbf{g}\right) \mathbf{p}. \tag{25}$$

The shrinkage functions are jointly learned by solving the following LS problem

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \left\|\mathbf{y} - \hat{\mathbf{y}}\left(\mathbf{g}, \mathbf{p}\right)\right\|_{\mathbf{2}}^{\mathbf{2}} \tag{26}$$

and the solution is given by

$$\hat{\mathbf{p}} = \left(\mathbf{L}^{\mathbf{T}}\mathbf{L}\right)^{-\mathbf{1}} \mathbf{L}^{\mathbf{T}}\mathbf{y}, \tag{27}$$

where $\mathbf{L} = \mathbf{L}\left(\mathbf{g}\right)$.

4.2 Stage II: Super-Resolution Reconstruction

Once the parameters of the shrinkage functions are learned, the super-resolution image is reconstructed as follows

$$\hat{\mathbf{x}}\left(\mathbf{y}_{\uparrow}, \hat{\mathbf{p}}\right) = \left(B^{T}B\right)^{-1} \sum_{k=1}^{K} B_{k}^{T} S_{\mathbf{q}_k}\left(\mathbf{y}_{\uparrow_{B_k}}\right) \hat{\mathbf{p}}_k. \tag{28}$$

5 Performance Evaluation

The performance of the proposed algorithm was evaluated by computer simulations and compared versus bi-cubic interpolation and the state-of-the-art sparse-representation based algorithm [1], [2] (which outperforms the sparse-representation approach [3]). Performance were not compared to the method [5]

Fig. 2. The collection of tested images, all images are of size 512×512

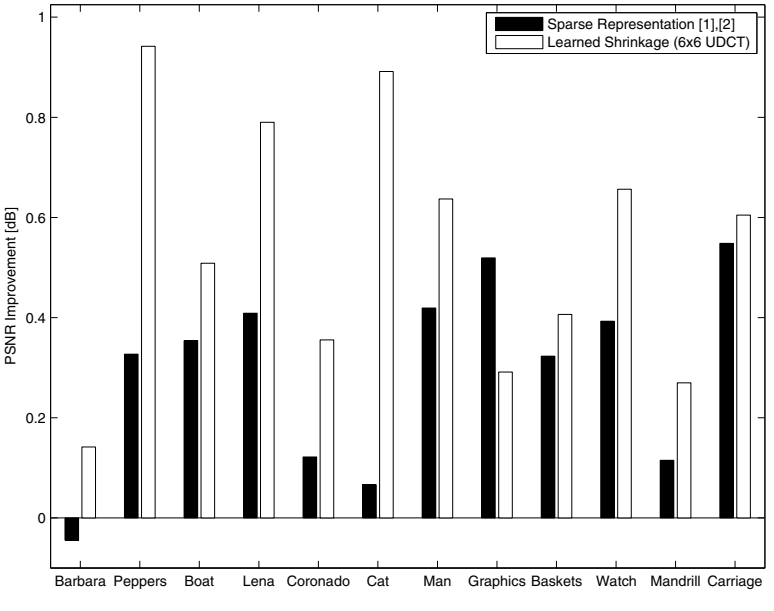


Fig. 3. PSNR improvement over bi-cubic interpolation for $L = 9$

Table 1. PSNR results for $L = 9$

Image	Bicubic Interp.	Sparse Rep. [1]	Shrink. 4×4	Shrink. 6×6	Shrink. 8×8	Offline Shrink.
Barbara	24.05	24.00	24.18	**24.19**	24.17	24.06
Peppers	29.82	30.14	30.51	**30.76**	30.75	29.77
Boat	27.14	27.49	27.58	**27.65**	27.61	27.31
Lena	30.76	31.17	31.45	**31.55**	31.53	30.85
Coronado	25.24	25.36	25.53	**25.59**	25.46	25.36
Cat	28.67	28.73	29.32	29.56	**29.59**	28.79
Man	28.35	28.77	28.92	**28.98**	28.91	28.46
Graphics	21.79	22.31	**22.36**	22.08	22.29	21.94
Baskets	21.13	21.45	21.48	**21.54**	21.51	21.20
Watch	28.10	28.49	28.60	**28.75**	28.68	28.18
Mandrill	22.01	22.13	22.24	22.28	22.27	**22.86**
Carriage	27.41	27.96	27.90	**28.02**	27.98	27.47

Fig. 4. Super-resolution of the image *barbara* for $L = 4$

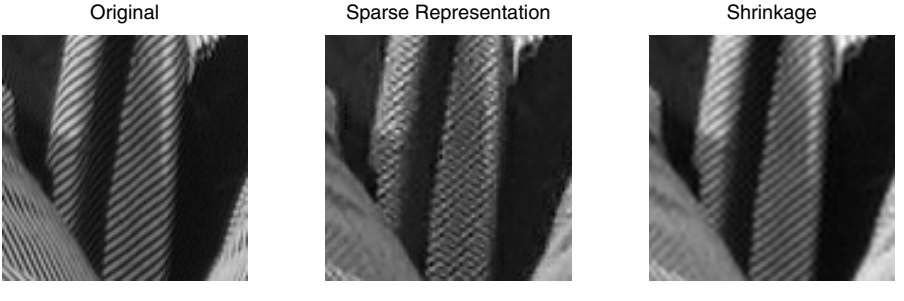


Fig. 5. Super-resolution of the image *barbara* for $L = 4$

Fig. 6. Super-resolution of the image *watch* for $L = 4$

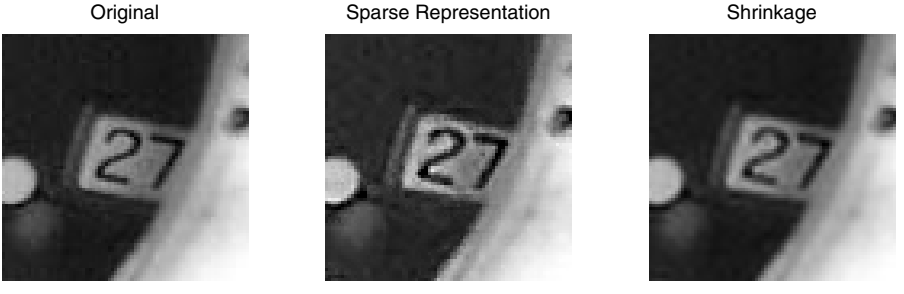


Fig. 7. Super-resolution of the image *watch* for $L = 4$

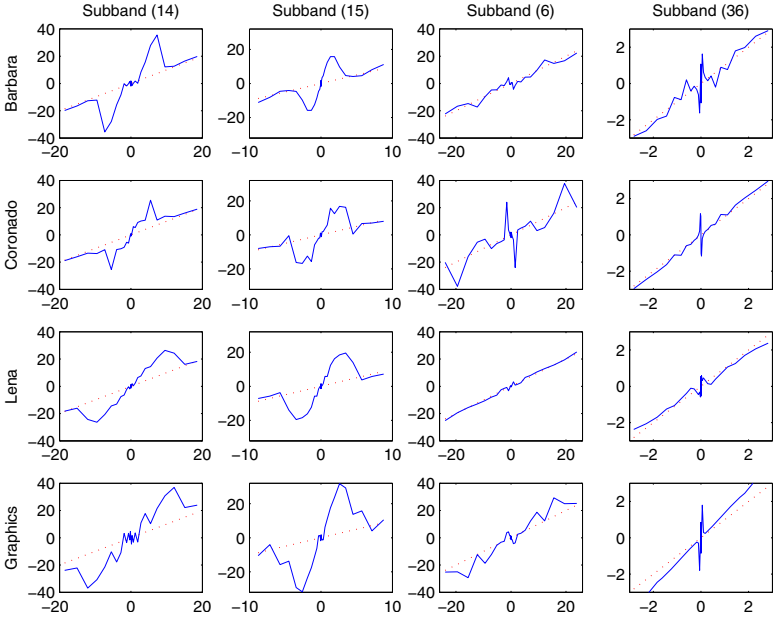


Fig. 8. Examples of learned shrinkage functions for $L = 9$. In each row the image is fixed and in each column the subband is fixed.

as neither quantitative results nor a code of this method were available for evaluation. A collection of 12 images presented in Fig. 2 were compared against their own SISR reconstructions (from their scaled-down versions) with scale-up factors of $L = 4$ and $L = 9$. The undecimated $\sqrt{K} \times \sqrt{K}$ windowed DCT (UDCT) was chosen as the overcomplete transform. This transform is defined to include all possible $\sqrt{K} \times \sqrt{K} = K$ DCT window shifts, leading to a redundancy factor of K with K distinct subbands. In this approach, each subband $\mathbf{y}_{\uparrow_{B_k}} = B_k \mathbf{y}_{\uparrow}$ is generated by filtering the image with the respective basis kernel. In addition, the UDCT is a *tight frame* thus the term $\left(B^T B\right)^{-1}$ boils down to the identity matrix. PSNR results are compared in Fig. 3 for $L = 9$ and a 6×6 UDCT, where it can be seen that the proposed approach outperforms the sparse-representation approach for all the images (excluding the image *graphics*). The proposed approach achieved an average gain of 0.54dB over bi-cubic interpolation, versus an average gain of 0.30dB achieved by the sparse-representation approach. Detailed PSNR results are presented in Table 1 for all methods. Three different UDCT window sizes were compared for online learning and it can be seen that the 6×6 window size provided the best results. In addition, the offline learning approach [9] was evaluated by training the shrinkage functions with an image from the training collection reported in [9]. The offline training was performed using equation (24), with a 6×6 UDCT. It can be seen that the offline approach provided inferior results compared to the online approach (excluding the image *mandrill*). In the specific case of the image *graphics*, it is possible that the assumption of scale-invariance is not as true, explaining the lower performance obtained. These type of images could be treated using the offline approach with adequately chosen training examples. Visual comparison of SISR reconstructions are presented for $L = 4$ in Figs. 4–7, it can be seen that various artifacts appear in the sparse-representation based approach while the proposed approach produces more natural and pleasant results (figures are best viewed in the electronic version of this paper). Examples of learned shrinkage functions are presented in Fig. 8, where it can be seen that for a fixed subband the learned shrinkage functions exhibit significantly different behavior for different images. For instance, only in subband (6) of the image *coronado* there is significant boosting effect with sign inversion for low amplitude coefficients.

6 Conclusions

This paper presented a novel approach for shrinkage functions learning in single image super-resolution. By exploiting the scale-invariant property of natural images, the set of scalar shrinkage functions are jointly learned from the low-resolution input image. Computer simulations with a simple overcomplete dictionary - the undecimated windowed DCT - revealed superior performance versus the state-of-the-art sparse-representation approach. Future research directions include a joint online-offline learning approach that combines additional example images into the online learning process. In addition we will consider the reconstruction of the residual error image in (15) rather then the complete image - thus focusing the learning process only into the missing high-pass components.

Acknowledgement

This research was partly supported by the European Community's FP7-FET program, SMALL project, under grant agreement no. 225913, and by the ISF grant number 599/08.

References

1. Yang, J., Wright, J., Ma, Y., Huang, T.: Image super-resolution as sparse representation of raw image patches. In: CVPR (2008)
2. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution via sparse representation. Submitted to IEEE Trans. on Image Processing (2010)
3. Wang, J., Zhua, S., Gonga, Y.: Resolution enhancement based on learning the sparse association of image patches. Pattern Recognition Letters 31, 1–10 (2010)
4. Freeman, W.T., Jones, T.R., Pasztor, E.C.: Example-based super-resolution. IEEE Computer Graphics and Applications 22, 56–65 (2002)
5. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: ICCV (2009)
6. Elad, M., Datsenko, D.: Example-based regularization deployed to super-resolution reconstruction of a single image. The Computer Journal 50, 1–16 (2007)
7. Guleryuz, O.G.: Predicting wavelet coefficients over edges using estimates based on nonlinear approximants. In: Proceedings of the Data Compression Conference (2004)
8. Mueller, N., Lu, Y., Do, M.N.: Image interpolation using multiscale geometric representations. In: SPIE Symposium on Electronic Imaging, San Jose (2007)
9. Hel-Or, Y., Shaked, D.: A discriminative approach for wavelet denoising. IEEE Trans. on Image Processing 17, 443–457 (2008)
10. Ruderman, D.L., Bialek, W.: Statistics of natural images: Scaling in the woods. Phys. Rev. Lett. 73, 814–817 (1994)
11. Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381, 607–609 (1996)
12. Elad, M.: Why simple shrinkage is still relevant for redundant representations? IEEE Trans. on Information Theory 52, 5559–5569 (2006)
13. Donoho, D.L., Johnstone, I.M.: Ideal spatial adaptation by wavelet shrinkage. Biometrika 81, 425–455 (1994)
14. Donoho, D.L.: Denoising by soft thresholding. IEEE Trans. on Information Theory 41, 613–627 (1995)
15. Coifman, R.R., Donoho, D.L.: Translation invariant de-noising. In: Lecture Notes in Statistics: Wavelets and Statistics, pp. 125–150 (1995)
16. Ma, J., Plonka, G.: The curvelet transform. IEEE Signal Processing Magazine 27, 118–133 (2010)
17. Do, M.N., Vetterli, M.: The contourlet transform: an efficient directional multiresolution image representation. IEEE Trans. on Image Processing 14, 2091–2106 (2005)
18. Guleryuz, O.G.: Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-part i: theory. IEEE Trans. on Image Processing 15, 539–554 (2006)

Object of Interest Detection by Saliency Learning

Pattaraporn Khuwuthyakorn[1,3], Antonio Robles-Kelly[1,2], and Jun Zhou[1,2]

[1] RSISE, Australian National University, Canberra, ACT 0200, Australia
[2] National ICT Australia (NICTA*), Canberra, ACT 2601, Australia
[3] Cooperative Research Centre for National Plant Biosecurity**, Canberra, ACT, 2617, Australia

Abstract. In this paper, we present a method for object of interest detection. This method is statistical in nature and hinges in a model which combines salient features using a mixture of linear support vector machines. It exploits a divide-and-conquer strategy by partitioning the feature space into sub-regions of linearly separable data-points. This yields a structured learning approach where we learn a linear support vector machine for each region, the mixture weights, and the combination parameters for each of the salient features at hand. Thus, the method learns the combination of salient features such that a mixture of classifiers can be used to recover objects of interest in the image. We illustrate the utility of the method by applying our algorithm to the MSRA Salient Object Database.

1 Introduction

Saliency map is an important tool in vision research [1]. Each pixel in this map is assigned with a measure of "relevance" or "importance" so as to reflect the degree to which a region in the image is attractive to visual attention. The research on visual saliency has generated a vast literature in computer vision and found applications in many areas, such as region of interest extraction [2], segmentation [3], tracking [4], object detection [5], thumbnailing [6] and image retrieval and classification [7].

It has been widely accepted that visual saliency computation can be effected in a bottom-up manner [8,9,10,11]. Departing from this strategy, Itti et al. [9] proposed a computational framework for visual saliency which decomposes visual input into component feature maps. In [12], Alter and Basri used image edges to construct the saliency map. The work in [12] is in line with the common approach to model contour or curve saliency, where length and smoothness of the edge points are often used [13,14].

* NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

** Pattaraporn Khuwuthyakorn would like to acknowledge the support the Australian Governments Cooperative Research Centres Program.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 636–649, 2010.

The combination of individual features into saliency maps can be greatly influenced by the behavioral goal of human attention [15]. This can be considered as a top-down modulation mechanism [16]. Note that, when guided by observer preferences, those parts that are less related to the visual targets of visual attention can be assigned smaller contributions on the saliency map or even completely ignored. To model this process, Navalpakkam and Itti [17] proposed a method to maximise the signal-to-noise ratio between the mean salience of the target and that of the distractor. Berengolts and Lindenbaum [14] also proposed a method to recover the distribution of the edge lengths and curvature on the region corresponding to the target of interest making use of labelled objects. In [18], saliency maps were computed as a linear combination of features whose weights were recovered through a linear regression model applied to manually labeled images. Liu et al. [2] formulated the saliency detection problem as a region of interest segmentation task where learning is performed via a conditional random field.

Note that, in some of the methods above, the same features at different scales are added together in a linear fashion [9,2] or modelled in a scale-space setting [19]. This suggest that salient objects or regions with different sizes may generate the same contribution to the final saliency map. Moreover, the intrinsic relationships between the individual features is often overlooked. This is due to the fact that, in existing methods, the optimisation step treats the features as independent primitives, despite the fact that they may actually be interrelated or highly correlated. This is even more important since, in the case of saliency features, we often deal with a large sample size with moderate feature dimension. Thus, for purposes of saliency learning, the features may span a space which is nonlinear in nature. This is in contrast with other settings in computer vision where linear classifiers can be applied on high dimensional features.

Hence, in this paper, we present a method which aims at combining salient features through a structured learning characterisation of the problem so as to achieve two desirable properties. Firstly, recovering a classifier model with the efficiency of linear Support Vector Machines. Secondly, reaching the discrimination power of nonlinear classifiers. To do this, we adopt a divide-and-conquer strategy that exploits partitioning the feature space into regions that are linearly separable. This is effected through a mixture of Support Vector Machines (SVMs) where the mixture weights and the feature combination coefficients are optimised using an Expectation-Maximisation (EM) approach. The method presented here is quite general in nature and can accommodate a number of saliency features found in the literature. In our work, we make use of the multi-scale features in [9] and [2], and present their natural extensions to neighbourhood-based descriptors.

2 Structured Learning

As mentioned earlier, our object of interest detection method makes use of saliency features and structured learning. The structured learning approach hinges in the notion that non-linear classification can be effected in a piecewise-linear manner

across the feature space. This provides a means to efficiency through the use of linear classifiers while preserving the flexibility of non-linear methods. Our probabilistic formulation employs two ingredients. The first one is the prior probability of the mixture given a feature-set at a pixel-site on the image. The second ingredient is the posterior probability corresponding to the outputs for each of the linear SVMs.

2.1 Mixture of SVMs

In this section, we cast the recovery of the saliency map into a structured learning setting. The aim is to combine the saliency features so as to perform classification, i.e. separate salient objects from the background in the image, based upon objects of interest provided as training data. Here, we formulate the problem in terms of a generative model over the training data. This joint distribution model enables us to explicitly incorporate mixture coefficients into the likelihood function. Consequently, we can perform parameter learning and model selection simultaneously by imposing a proper prior on the mixture co-efficients based on the minimum message length (MML) criterion [20]. Parameter update is then achieved making use of the EM algorithm [21]. For model selection, we start with an overcomplete model and automatically prune vanishing SVM mixture co-efficients. Hence structured learning is implicitly incorporated into the optimisation process and performed in a top-down manner.

To commence, consider a set of M tuples $(X,Y)=\{(\mathbf{x}_{i,l},y_i)|i=1,\ldots,M,y_i\in\{-1,1\}\}$, where $(\mathbf{x}_{i,l},y_i)$ are the i^{th} data-label pair in the training data corresponding to the l^{th} saliency feature, where the total number of salient feature is N. In practice, Y accounts for the corresponding object of interest regions provided at input. The linear SVM classifier solves the following optimisation problem

$$\min_{\mathbf{w}} \quad \frac{||\mathbf{w}||^2}{2} + C\sum_i \epsilon(\mathbf{w};\mathbf{x}_{i,l},y_i) \tag{1}$$

where $\epsilon(\mathbf{w};\mathbf{x}_{i,l},y_i)=\max\left(1-y_i\mathbf{w}^T\mathbf{x}_{i,l},0\right)$ is the Hinge loss function which specifies an upper bound on the classification error. The first term on the right hand side is regularisation term on classifier weights. Without loss of generality, we have subsumed the bias term b in the above formulation by appending each data instance with an additional dimension $\mathbf{x}_{i,l}^T=[\mathbf{x}_{i,l}^T,1]$ and $\mathbf{w}^T=[\mathbf{w}^T,b]$.

We can extend the SVM model above to a two-layer mixture model formulated using the joint probability distribution over the salient regions provided by the user and the SVM binary classifier. The model, hence, consists of two parts. The hidden layer, which is composed of the gating network that produces a soft-partition of the input space by generating a data-dependent weight distribution. Each node in the hidden layer is connected to a linear SVM classifier in the input layer, which is responsible for the salient object recovery.

We establish the link between the proposed mixture model and the associated generative model using the joint probabilistic distribution over the data in X and the labels in Y given by

$$P(Y|X,\Theta) = \prod_i P(y_i|\mathbf{x}_{i,l},\Theta) = \prod_i \sum_{z_i} P(y_i|z_i,\mathbf{x}_{i,l},\Theta)P(\mathbf{x}_{i,l}|z_i,\Theta)P(z_i \mid \Theta) \tag{2}$$

where i indexes data samples as before, $\Theta = \{\alpha, \beta, \tau, \gamma\}$ are the parameters of the underlying model and z_i is the hidden variable introduced for the ith sample for each of the N salient features under study. In the equation above, α and β are the multinomial parameters that generate the hidden variables z_i's whereas τ and γ are parameters for the gating nodes and classifiers, whose specific parametric forms will be explained later. The probability $P(\mathbf{x}_{i,l}|z_i,\tau)$ represents the posterior for the mixture component with hyperparameters τ, and $P(y_i|\mathbf{x}_{i,l},\gamma)$ is the posterior probability of corresponding linear SVM output for the ith sample.

It is worth noting that our mixture of SVMs model can also be viewed from the perspective of graphical model due to its generative nature. From this viewpoint, $\mathbf{x}_{i,l}$ and y_i are the target random variables whose joint distributions are to be modeled, and z_i is the hidden variable generated from a multinominal distribution with parameters $\alpha = \{\alpha_1, \ldots, \alpha_K\}$ and $\beta = \{\beta_1, \ldots, \beta_N\}$ for K-mixtures and N features. Thus, $\mathbf{x}_{i,l}$ is generated from an isotropic Gaussian distribution with parameter τ conditional on z_i, where $\tau = \{(\mu_{1,1}, \Sigma_{1,1}), \ldots, (\mu_{K,N}, \Sigma_{K,N})\}$ and $\mu_{j,l}$ and $\Sigma_{j,l}$ are the mean vector and the variance for the jth mixture component performing inference upon the saliency feature-set indexed l. The target random variable y_i is generated from a probabilistic classifier model with parameter γ conditional on $\mathbf{x}_{i,l}$ and z_i, where $\gamma = \{\mathbf{w}_{1,1}, \ldots, \mathbf{w}_{K,N}\}$, and $\mathbf{w}_{j,l}$ is the classifier weight-vector for the jth linear SVM corresponding to the l^{th} saliency feature-set. This yields

$$P(Y|X,\Theta) = \prod_i \sum_{z_i} P(y_i|\mathbf{x}_{i,l},\gamma)P(\mathbf{x}_{i,l}|z_i,\tau)P(z_i \mid \alpha,\beta) \tag{3}$$

The proposed model bears some resemblance with the mixture of experts (HME) model proposed by Jacobs and Jordan [22]. Nonetheless, they are inherently different in nature in the sense of the probabilistic distributions they capture. Our model captures the joint distribution of data and labels, whereas the HME model is associated with the conditional probability distribution of labels given the data. In the HME model, the hidden variable z_i is generated from a conditional probability distribution while in our method it arises from a multinominal distribution with parameter α. This enables us to control the complexity of the model implicitly by enforcing proper sparseness priors on α.

Equation 2 suggests parameter estimation can be effected via Maximum Likelihood Estimation (MLE) by maximising the following log-likelihood function

$$\begin{aligned}\mathcal{L}(\Theta) &= \sum_i \log P(y_i|\mathbf{x}_{i,l},\Theta) + \sum_j \Omega(\mathbf{w}_{j,l}) \\ &= \sum_i \log\Big\{\sum_l \beta_l \sum_j \alpha_j P(y_i|\mathbf{x}_{i,l},\mathbf{w}_{j,l})P(\mathbf{x}_{i,l}|z_i,\tau)\Big\} + \sum_j \Omega(\mathbf{w}_{j,l})\end{aligned} \tag{4}$$

where $\Omega(\mathbf{w}_{j,l}) = \log\{P(\mathbf{w}_{j,l})\}$ is a log-prior term for regularisation purposes. The last line follows from Equation 3, the definition of $\gamma = \{\mathbf{w}_{1,1}, \ldots, \mathbf{w}_{K,N}\}$ and the use of the shorthand $P(z_i \mid \alpha, \beta) = \alpha_j \beta_l$ for the j^{th} mixture and the l^{th} salient feature-set. This responds to the fact that here, we view $P(z_i \mid \alpha, \beta)$ as a data-independent term which specifies the prior probability of the mixture and salient feature pair at a given pixel-site on the image.

In order to incorporate the linear SVM into the log-likelihood above, we view the associated constrained quadratic optimisation problem corresponding to the negative log-likelihood from a probabilistic viewpoint. Note that the second term on the right hand side is related to the prior $\Omega(\mathbf{w})$, whereas the first term corresponds to the conditional probability $P(y|\mathbf{x}, \mathbf{w})$ related to classification errors. These are given by

$$\Omega(\mathbf{w}_{j,l}) = -\zeta ||\mathbf{w}_{j,l}||^2 \tag{5}$$

$$P(y_i|\mathbf{x}_{i,l}, \mathbf{w}_{j,l}) = e^{-\epsilon(\mathbf{w}_{j,l}; \mathbf{x}_{i,l}, y_i)} \tag{6}$$

Here we have omitted the normalisation factor for the conditional probability $P(y_i|\mathbf{x}_{i,l}, \mathbf{w}_{j,l})$, which leads to an approximation of the probability measure. This is mainly due to the consideration regarding the use of numerical optimisation which enables us to employ existing fast linear SVM solvers [23] for parameter estimation. This simplification is still valid in the large margin case where the probability of the negative class is usually very small. More importantly, the likelihood function in Equation 4 is guaranteed to increase using the EM algorithm, as we discuss in the next section, regardless of whether or not $P(y_i|\mathbf{x}_{i,l}, \mathbf{w}_{j,l})$ is a proper probability measure over y_i.

2.2 The EM Algorithm

In this section, we describe an EM algorithm for solving the mixture of linear SVMs presented in the previous section. The E-step updates the posterior probability of assigning each sample to the component classifiers. Let $\Theta^{(t)} = \{\alpha_j^{(t)}, \beta_l^{(t)}, \mu_{j,l}^{(t)}, \Sigma_{j,l}^{(t)}, \mathbf{w}_{j,l}^{(t)} | j = 1, \ldots, K; l = 1, \ldots, N\}$ be the parameters at the current iteration, the probability of the ith sample given the j^{th} classifier and the l^{th} saliency feature is given by

$$q_{i,j,l}^{(t+1)} = \frac{\alpha_j^{(t)} \beta_l^{(t)} P(\mathbf{x}_{i,l}|\mu_{j,l}^{(t)}, \Sigma_{j,l}^{(t)}) P(y_i|\mathbf{x}_{i,l}, \mathbf{w}_{j,l}^{(t)})}{\sum_s \sum_u \sum_v \alpha_u^{(t)} \beta_v^{(t)} P(\mathbf{x}_{s,v}|\mu_{u,v}^{(t)}, \Sigma_{u,v}^{(t)}) P(y_s|\mathbf{x}_{s,v}, \mathbf{w}_u^{(t)})} \tag{7}$$

where $s \in \{1, \ldots, M\}$, $u \in \{1, \ldots, K\}$, $v \in \{1, \ldots, N\}$. $P(y_i|\mathbf{x}_{i,l}, \mathbf{w}_{j,l}^{(t)})$ is given by Equation 6, and $P(\mathbf{x}_{i,l}|\mu_{j,l}^{(t)}, \Sigma_{j,l}^{(t)})$ is given by the following multivariate, d-dimensional Gaussian distribution,

$$P(\mathbf{x}_{i,l}|\mu_{j,l}^{(t)}, \Sigma_{j,l}^{(t)}) = \frac{1}{\sqrt{(2\pi)^d \mid \Sigma_{j,l}^{(t)} \mid}} \exp\left(-\frac{1}{2}(\mathbf{x}_{i,l} - \mu_{j,l}^{(t)})^T \left(\Sigma_{j,l}^{(t)}\right)^{-1} (\mathbf{x}_{i,l} - \mu_{j,l}^{(t)})\right) \tag{8}$$

The M-step involves simultaneously updating the parameters for the gating nodes and SVM classifiers so as to solve two independent optimisation problems. Parameter estimation for the gating nodes is similar to the estimation of parameters for the Gaussian mixture model. Specifically, for the jth mixture component and lth saliency feature we have

$$\alpha_j^{(t+1)} = \frac{\sum_s \sum_v q_{s,j,v}^{(t+1)}}{\sum_s \sum_u \sum_v q_{s,u,v}^{(t+1)}} \tag{9}$$

$$\beta_l^{(t+1)} = \frac{\sum_s \sum_u q_{s,u,l}^{(t+1)}}{\sum_s \sum_u \sum_v q_{s,u,v}^{(t+1)}} \tag{10}$$

$$\mu_{j,l}^{(t+1)} = \frac{\sum_s q_{s,j,l}^{(t+1)} \mathbf{x}_{s,l}}{\sum_s q_{s,j,l}^{(t+1)}} \tag{11}$$

$$\Sigma_{j,l}^{(t+1)} = \frac{\sum_s q_{s,j,l}^{(t+1)} (\mathbf{x}_{s,l} - \mu_{j,l}^{(t+1)})^T (\mathbf{x}_{s,l} - \mu_{j,l}^{(t+1)})}{\sum_s q_{s,j,l}^{(t+1)}} \tag{12}$$

As a result, parameter estimation for the linear SVMs reduces itself to updating the classifiers for reweighted samples where the weights are specified by the posterior probabilities computed in the E-step. Specifically, for the j^{th} linear classifier working on the l^{th} saliency feature we solve the following classification problem

$$\max \sum_i \sum_l q_{i,j,l}^{(t)} \log P(y_i | \mathbf{x}_{i,l}, \theta_{j,l}) + \log P(\theta_{j,l}) \tag{13}$$

$$= \max \left\{ -\sum_i \sum_l q_{i,j,l}^{(t)} \epsilon(\mathbf{w}_{j,l}; \mathbf{x}_{i,l}, y_i) - \zeta ||\mathbf{w}_{j,l}||^2 \right\}$$

where $\theta_{j,l} = \{\alpha_j, \beta_l, \mu_{j,l}, \Sigma_{j,l}, \mathbf{w}_{j,l}\}$ and $C = \frac{1}{2\zeta}$. This is exactly the same problem as training linear SVMs in Equation 1 whose sample weights are given by $q_{i,j,l}^{(t)}$.

2.3 Convergence

As mentioned in the sections above, the method proceeds in an iterative fashion. At each iteration t, the method comprises the following steps

- Train the SVMs using the sample weights $q_{i,j,l}^t$ so as to recover the probabilities $P(y_i \mid \mathbf{x}_{i,l}, \mathbf{w}_{j,l}^{(t)})$. In practice, this is equivalent to obtaining the probabilistic output of the SVM classifiers as shown in [24].
- With $P(y_i \mid \mathbf{x}_{i,l}, \mathbf{w}_{j,l}^{(t)})$ at hand, compute the updated weights $q_{i,j,l}^{t+1}$ in Equation 7. These can be computed making use of the probabilities $P(\mathbf{x}_{i,l} \mid \mu_{j,l}^{(t)}, \Sigma_{j,l}^{(t)})$ given in Equation 8 and the probabilities $P(y_i \mid \mathbf{x}_{i,l}, \mathbf{w}_{j,l}^{(t)})$ recovered in the previous step.
- Recover the remaining parameters making use of Equations 9-12.

It should be noted that each EM iteration increases the log-likelihood given by Equation 4. This argument can be easily established by making use of the auxiliary function parameterised with respect to $\Theta^{(t)}$ given by

$$Q(\Theta;\Theta^{(t)}) = \sum_{i,j,l} q_{i,j,l}^{(t)} \log \alpha_j \log \beta_i P(\mathbf{x}_{i,l}|\mu_{j,l},\Sigma_{j,l}) P(y_i|\mathbf{x}_{i,l},\mathbf{w}_{j,l}) - \sum_i \sum_j \sum_l q_{i,j,l}^{(t)} \log q_{i,j,l}^{(t)} + \sum_j \Omega(\mathbf{w}_{j,l}) \quad (14)$$

which is the lower bound of $\mathcal{L}(\Theta)$ since

$$\mathcal{L}(\Theta) - Q(\Theta,\Theta^{(t)}) = q_{i,j,l}^{(t)} \log \frac{q_{i,j,l}^{(t)}}{q_{i,j,l}} \quad (15)$$

The gap is non-negative and varnishes if and only if $\Theta = \Theta^{(t)}$. Hence, the log-likelihood increases with the following relation

$$\mathcal{L}(\Theta^{(t+1)}) \geq Q(\Theta^{(t+1)},\Theta^{(t)}) \geq Q(\Theta^{(t)},\Theta^{(t)}) = \mathcal{L}(\Theta^{(t)})$$

The second inequality is true due to the maximisation step. Therefore, by repeating the EM steps we can obtain a convergent solution of the original maximum likelihood estimation problem. Moreover, we can stop the iteration presented earlier when the quantity $||\Theta^{(t+1)} - \Theta^{(t)}||$ is less or equal to a predefined threshold ρ.

3 Feature Extraction

So far, we have assumed the saliency features are at hand as input to our mixture of linear SVMs. Here, we elaborate further on the saliency features used in our experiments. It is worth noting that the developments above are general in nature and can be applied to a large variety of saliency features. Here, we depart from the feature map extraction methods by Itti et al. [9] and Liu et al. [2]. We extend these two methods by considering the pixel neighbourhood, which permits capturing the image structure during the feature extraction process. The individual features are then used as the input to our structured learning method.

In the Salient Map (SM) method of Itti et al. [9], an input image is first smoothed using Gaussian filters so as to generate a scale pyramid. Simple features are then extracted at each scale to generate three types of visual cues. The first of these is the intensity feature obtained by averaging the red, green and blue channel-values at each pixel in the input image. By computing the differences between seven scales, 6 intensity channels are recovered. The second set of features is based upon color and simulate the function of the cortex, which is represented by a set of color opponency between red, green and blue channel values against the yellow basis. For each set of colour features, differences are recovered over three scales and, hence, yield 12 channels. The third set is comprised by orientation features, which are given by the responses of a set of even-symmetric Gabor filters [25]. In practice, these are treated as a Gaussian envelope modulated by a complex sinusoidal carrier. Here, we compute the responses at six scales and four orientations, and thus, recover 24 orientation channels.

The method from Liu et al. [2], which we denote LRG, recovers saliency making use of local, regional and global features. The first of these consists of the local feature extracted from multi-scale contrast. For a given pixel, the image contrast is computed as the sum of the 2-norm grayscale differences between a pixel and its neighborhood. Then, contrast at different scales is combined linearly. To extract the regional salient feature-set, two bounding boxes are used. These cover the proposed salient object and its surrounding area. The differences between the RGB color histograms for the bounding boxes are computed so as to find the optimal center-surround aspect ratio of the object. Finally, the global saliency features are computed from spatial color distributions. This feature can be viewed as that represented by spatial color clusters, where colors with small spatial variance are assigned higher salience.

Despite effective, the features above may be prone to corruption due to noise and cluttered background. Furthermore, small objects may generate scattered salient regions during the feature extraction process. These greatly influence the final object of interest detection step. To solve these problems, we extend the above mentioned features to a neighbourhood-based descriptor setting by considering the interaction of image pixels with the neighboring pixels. Here, we adopt a second-order Markov setting, that is, including the saliency features of the pixels in a 3×3 neighborhood. In this way, we can generate a descriptor at each pixel that contains saliency features from both the pixel itself and its neighborhood. It can be seem in the later experiments that such extension helps maintain the local consistency in the object of interest detection.

4 Experiments

We perform experiments on the Microsoft Research Asia (MSRA) Salient Object Database B, which contains 5,000 images. Details on this database can be found in [2]. Our motivation in using this dataset stems in providing results consistent to those reported in [2] and, thus, presenting a fair comparison with the alternatives reported in the literature. We have randomly divided the images in the database into two groups of 2,500 images each. One of these is used for training and the other one for testing. At training, we set the number of SVMs for our mixture to five, i.e. $K = 5$. The SVM parameters have been recovered by ten-fold cross-validation. For our experiments, we have used four sets of features. The first set is the colour, contrast and center-surround features in [2] (LRG), thus, $N = 3$. The second set comprises the 42 channels generated from orientation, intensity and colour features in [9](SM). In this case, $N = 42$. We have also used the extensions of the features in [9] and [2] with a 3×3 neighbourhood $\mathcal{N}$ about each pixel in the imagery, which we denote SM-$\mathcal{N}$ with $N = 42$ and LRG-$\mathcal{N}$ with $N = 3$, respectively.

To compare the learning performance of our mixture of linear SVMs (MLSVM) with alternatives elsewhere in the literature, we also provide results yielded by the Conditional Random Field (CRF) inference algorithm in [2] and the boosting algorithm ADABOOST$_{REG}$ in [26]. For the CRF algorithm, we have used the parameters in [2], whereas for the ADABOOST$_{REG}$ we have used 10 weak learners

with ten-fold cross validation so as to obtain the best set of parameters. For our method, we have set the stoping threshold ρ for the EM iteration to 0.001 and initialised the parameters in Θ as follows. The weights $\alpha_j^{(0)}$ are set to $\frac{1}{K}$, i.e. $\alpha_j^{(0)} = \frac{1}{5}$. Similarly, we have set the feature weights to $\frac{1}{N}$, which yields the value for $\beta_j^{(0)}$. The means $\mu_{j,l}^{(0)}$ and covariances $\Sigma_{j,l}^{(0)}$ have been computed via k-means clustering [27]. To do this, we set $k = 5$ and apply k-means to each of the feature-sets under study. With the cluster members at hand, the corresponding means and covariances are computed.

For purposes of testing, we used the trained model to generate saliency values for each pixel. For the three methods, i.e. our approach, the CRF and the ADABOOST$_{REG}$, the testing output is a saliency map which indicates the probability of a testing pixel being the salient object. To detect a salient object region, we apply the optimal threshold recovery method in [28] on the saliency map. Following [2], we assume that there is only one salient object per image. Here, we extract the region whose size is largest amongst those yielded after the method in [28] is applied. Note that such setting is for the sake of providing an equal comparison with results reported elsewhere rather than a limitation on our method. More than one objects may be obtained by sequentially extracting regions in order of their sizes.

To commence, we show sample results yielded by the 12 classifier-feature pairs used in our experiments (three learning methods against four feature sets). Figure 1 shows some examples of saliency maps recovered by our method and the alternatives for the images on the top-most row. The recovered objects of interest for the images shown in Figure 1 are shown in Figure 2. In the panels, the bounding boxes show the recovered regions after the application of the method in [28] to the saliency maps. Note that, despite the LRG-$\mathcal{N}$ features with the CRF inference produces results comparable to our approach, our method provides bounding boxes more in accordance with the ground truth. This is particularly evident for the coloured wine glasses and the tulip images. Moreover, for other images, such as the log-cabin and the CPU images, the LRG-$\mathcal{N}$ features with the CRF has slightly cropped the objects of interest by delivering smaller bounding boxes.

We now provide a quantitative analysis using a number of performance measures. The first of these is the precision-recall measure in [2]. The precision-recall formulation in [2] takes into account the structure of the database in our experiments by using the binary masks provided as ground truth and the ones delivered by our method and the alternatives. The second of the quantitative measures used here is the F-score [29]. The F-score is defined as $F_\eta = \frac{(1+\eta) precision \times recall}{\eta \times precision + recall}$. Following [30], we have set $\eta = 0.5$, which corresponds to the weighted harmonic mean of precision-recall. Finally, we have used the boundary displacement error (BDE) [31]. In our experiments, we have followed [2] and used the fixation area so as to compute our F-score and BDE plots. The fixation area is the smallest rectangle containing a fixed percentage of salient pixels as delivered by our method and the alternatives. As in [2], and so as to provide consistent results to

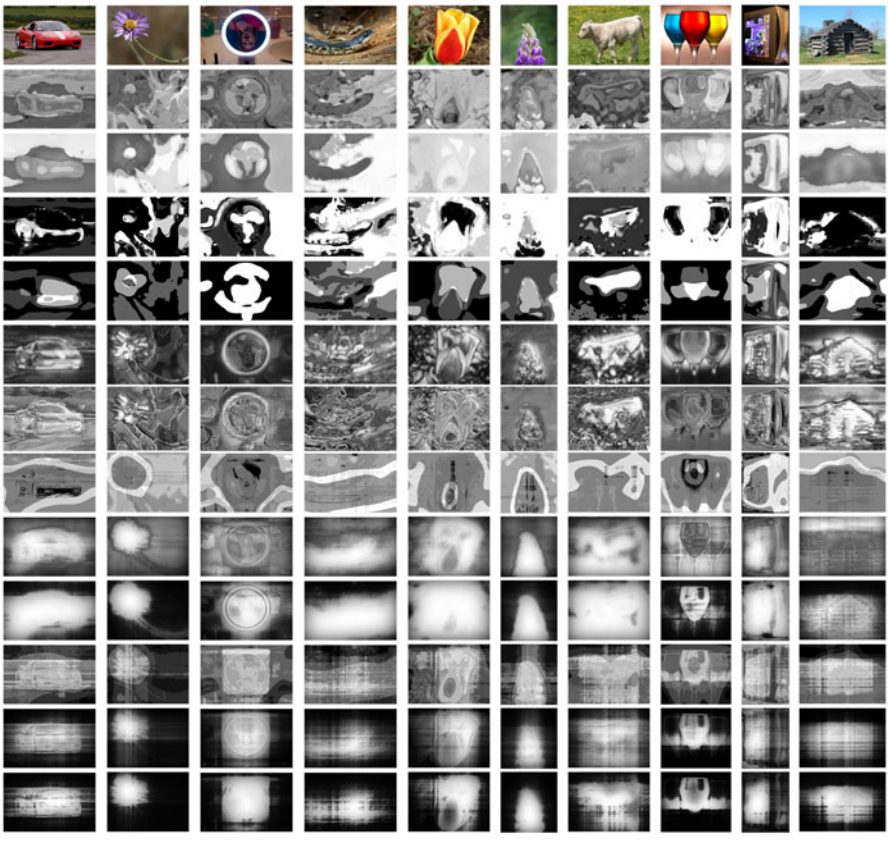


Fig. 1. Saliency map samples computed using different features and learning methods. From top-to-bottom: Ground truth, SM+ADABOOST$_{REG}$, SM+CRF, SM+MLSVM, SM-$\mathcal{N}$+ADABOOST$_{REG}$, SM-$\mathcal{N}$+CRF, SM-$\mathcal{N}$+MLSVM, LRG+ADABOOST$_{REG}$, LRG+CRF, LRG+MLSVM, LRG-$\mathcal{N}$+ADABOOST$_{REG}$, LRG-$\mathcal{N}$+CRF, LRG-$\mathcal{N}$+MLSVM.

those reported elsewhere, the fixation area has been recovered through exhaustive search.

In Figure 3 we show the overall dataset-average precision-recall plots for the 12 combinations of saliency feature-sets and inference methods used in our experiments. In the figure, for the sake of clarity, we have divided the plots into two panels. On the left-hand-side, we show those plots corresponding to the SM and SM-$\mathcal{N}$ features, whereas the other panels shows the results for the LRG and LRG-$\mathcal{N}$ features. Note that our method (MLSVM) performs best with both, the SM-$\mathcal{N}$ and the LRG-$\mathcal{N}$ features followed by the CRF with LRG-$\mathcal{N}$ features and the ADABOOST$_{REG}$ taking LRG-$\mathcal{N}$ features as input. Note that the varying length of the traces in the plot corresponds to the dependence of the precision-recall measurements upon the fixation area. In our plots, each of the markers

Fig. 2. Sample object of interest detection results. From top-to-bottom: Ground truth, SM+ADABOOST$_{REG}$, SM+CRF, SM+MLSVM, SM-$\mathcal{N}$+ADABOOST$_{REG}$, SM-$\mathcal{N}$+CRF, SM-$\mathcal{N}$+MLSVM, LRG+ADABOOST$_{REG}$, LRG+CRF, LRG+MLSVM, LRG-$\mathcal{N}$+ADABOOST$_{REG}$, LRG-$\mathcal{N}$+CRF, LRG-$\mathcal{N}$+MLSVM.

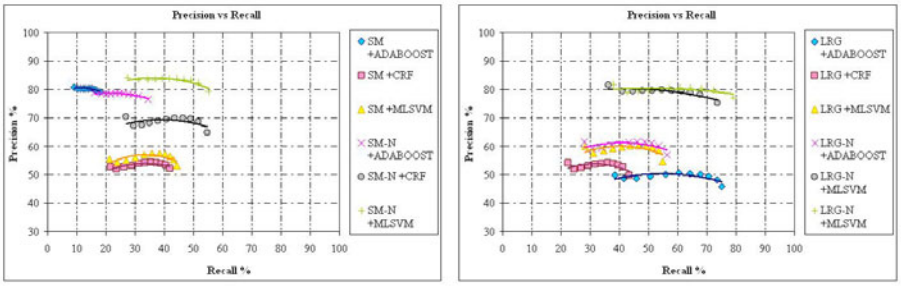


Fig. 3. Average precision-recall

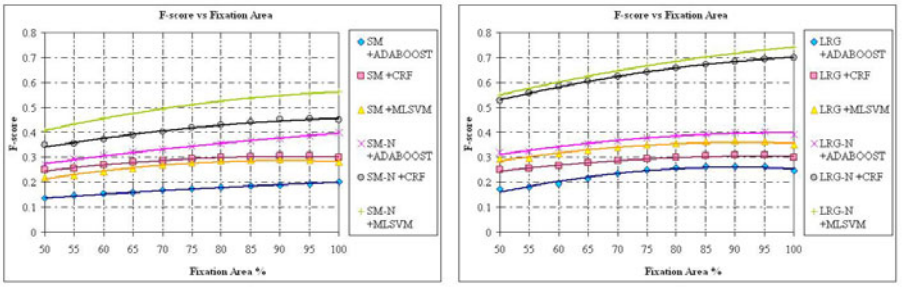


Fig. 4. Average F-score as a function of the fixation area percentage

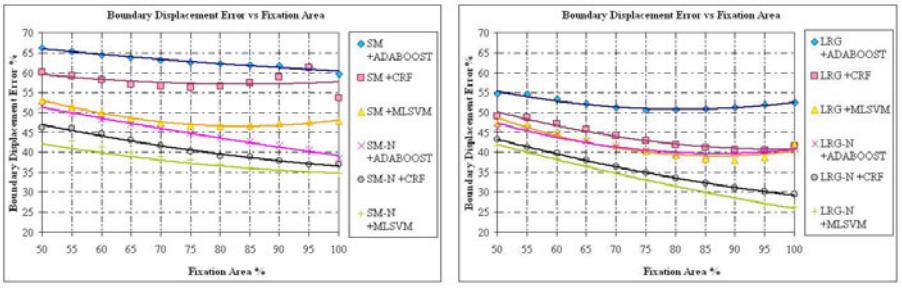


Fig. 5. Boundary Displacement Error as a function of the fixation area percentage

corresponds to fixation area variations from 50% to 100% in increments of 5%. As a result, the "flatter" and higher the precision-recall traces in the plot the more stable the classifier-feature pair is to variations of fixation area.

Following the observation that our measures are dependent on fixation area percentages, in Figures 4 and 5 we show the F-scores and BDE as a function of fixation area percentage. As in Figure 3, we have plotted, on the left-hand panels, the traces for the SM and SM-$\mathcal{N}$ features, while the right-hand plots correspond to the *LRG* and LRG-$\mathcal{N}$ feature-sets. On both figures, the neighbourhood-based saliency descriptors are always the best performers, regardless of the inference method used. In both accounts, the MLSVM with LRG-$\mathcal{N}$ features outperforms the alternatives, with lower BDEs and higher F-scores across the fixation area percentages, with ADABOOST$_{REG}$ consistently delivering the worst results. It is also worth nothing that the LRG based features shows better F-score and BDE results than SM based features. This is consistent with Figures 1, where the topmost six rows, corresponding to the results yielded using the SM and SM-$\mathcal{N}$ features, show regions which are less well defined than the panels in the bottom rows. The notion that the LRG and LRG-$\mathcal{N}$ features provide better performance is confirmed by the F-score results. Nonetheless, for all the quantitative

measures in our experiments, the MLSVM provided a margin of advantage over the alternative learning methods.

5 Conclusions

In this paper, we have presented a mixture of Linear SVMs for purposes of learning how to detect a salient object. The method presented here employs a mixture of linear SVMs so as to partition the feature space into sub-regions which are linearly separable. This is a divide-and-conquer approach which allows the recovery of the mixture weights and the feature combination coefficients making use of the EM algorithm. We have illustrated the utility of the method for purposes of recovering objects of interest in the MSRA Salient Object Database and compared our results to a number of alternatives. We have also provided neighbourhood-based descriptor extensions to the features presented in [2] and [9]. Note that the proposed method is quite general and can be applied to many other types of features which, in contrast with those used here, may not be local in nature.

References

1. Fecteau, J., Munoz, D.: Salience, relevance, and firing: a priority map for target selection. Trends in Cognitive Sciences 10, 282–290 (2006)
2. Liu, T., Sun, J., Zheng, N.N., Tang, X., Shum, H.Y.: Learning to detect a salient object. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
3. Mahamud, S., Williams, L., Thornber, K., Xu, K.: Segmentation of multiple salient closed contours from real images. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 433–444 (2003)
4. Li, H., Ngan, K.N.: Saliency model-based face segmentation and tracking in head-and-shoulder video sequences. Journal of Visual Communication and Image Representation 19, 320–333 (2008)
5. Papageorgiou, C., Poggio, T.: A trainable system for object detection. International Journal of Computer Vision 38, 15–33 (2004)
6. Marchesotti, L., Cifarelli, C., Csurka, G.: A framework for visual saliency detection with applications to image thumbnailing. In: Proceedings of the IEEE International Conference on Computer Vision (2009)
7. Kadir, T., Brady, M.: Saliency, scale and image description. International Journal of Computer Vision 45, 83–105 (2001)
8. Koch, C., Ullman, S.: Shifts in selective visual attention: Towards the underlying neural circuitry. Human Neurobiology 4, 219–227 (1985)
9. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 20, 1254–1259 (1998)
10. Harel, J., Koch, C., Perona, P.: Graph-based visual saliency. In: Proceedings of Neural Information Processing Systems, pp. 545–552 (2007)
11. Rosin, P.L.: A simple method for detecting salient regions. Pattern Recognition 42, 2363–2371 (2009)

12. Alter, T., Basri, R.: Extracting salient curves from images: An analysis of the saliency network. International Journal of Computer Vision 27, 51–69 (1998)
13. Shaashua, A., Ullman, S.: Structural saliency: The detection of globally salient structures using locally connected network. In: Proceedings of International Conference on Computer Vision, pp. 321–327 (1988)
14. Berengolts, A., Lindenbaum, M.: On the distribution of saliency. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 1973–1990 (2006)
15. Dickinson, S.J., Christensen, H.I., Tsotsos, J.K., Olofsson, G.: Active object recognition integrating attention and viewpoint control. Computer Vision and Image Understanding 67, 239–260 (1997)
16. Navalpakkam, V., Itti, L.: Modeling the influence of task on attention. Vision Research 45, 205–231 (2005)
17. Navalpakkam, V., Itti, L.: Search goal tunes visual features optimally. Neuron 53, 605–617 (2007)
18. Vincent, B., Troscianko, T., Gilchrist, I.: Investigating a space-variant weighted salience account of visual selection. Vision Research 47, 1809–1820 (2007)
19. Lindeberg, T.: Scale-space behaviour of local extrema and blobs. Journal of Mathematical Imaging and Vision 1, 65–99 (1992)
20. Rissanen, J.: Stochastic Complexity in Statistical Inquiry Theory. World Scientific Publishing Co., Inc., River Edge (1989)
21. Dempster, A.P., Laird, M.N., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 39, 1–22 (1977)
22. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Computation 3, 79–87 (1991)
23. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. The Journal of Machine Learning Research 9, 1871–1874 (2008)
24. Platt, J.: Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In: Advances in Large Margin Classifiers, pp. 61–74 (2000)
25. Daugman, J.: Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two dimensional visual cortical filters. Journal of the Optical Society of America 2, 1160–1169 (1985)
26. Ratsch, G., Onoda, T., Muller, K.R.: Soft margins for adaboost. Machine Learning 42, 287–320 (2001)
27. Duda, R.O., Hart, P.E.: Pattern Classification. Wiley, Chichester (2000)
28. Otsu, N.: A thresholding selection method from gray-level histobrams. IEEE Transactions on Systems, Man, and Cybernetics 9, 62–66 (1979)
29. van Rijsbergen, C.J.: Information Retireval. Butterworths (1979)
30. Martin, D.R., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 530–549 (2004)
31. Freixenet, J., Munoz, X., Raba, D., Martí, J., Cufí, X.: Yet another survey on image segmentation: Region and boundary information integration. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 408–422. Springer, Heidelberg (2002)

Boundary Detection Using F-Measure-, Filter- and Feature- (F^3) Boost

Iasonas Kokkinos

Department of Applied Mathematics, Ecole Centrale Paris
INRIA-Saclay, GALEN Group

Abstract. In this work we propose a boosting-based approach to boundary detection that advances the current state-of-the-art. To achieve this we introduce the following novel ideas: (a) we use a training criterion that approximates the F-measure of the classifier, instead of the exponential loss that is commonly used in boosting. We optimize this criterion using Anyboost. (b) We deal with the ambiguous information about orientation of the boundary in the annotation by treating it as a hidden variable, and train our classifier using Multiple-Instance Learning. (c) We adapt the Filterboost approach of [1] to leverage information from the whole training set to train our classifier, instead of using a fixed subset of points. (d) We extract discriminative features from appearance descriptors that are computed densely over the image. We demonstrate the performance of our approach on the Berkeley Segmentation Benchmark.

1 Introduction

The abundant biological evidence that our visual system employs sophisticated boundary detection mechanisms, and the legacy of D. Marr [2] has led early computer vision researchers to pursue computational approaches to boundary detection, considering it as the starting point for any subsequent processing. Moreover, the striking ease with which we recognize shape-based classes, e.g. sketches while being bereft of all appearance information also suggests that boundary detection may be the missing piece in the current, appearance-dominated, object recognition research.

A revival of research on boundary detection has been observed during the last years, largely due to the introduction of ground-truth labeled datasets [3,4] which facilitated the treatment of the problem in a machine learning framework, while weeding out many of the heuristics previously used in edge detection. Based on the consistent improvements observed during the last years on these benchmarks [5,6,7,8,9,10], boundary detection is anticipated to become an indispensable part of any computer vision 'toolbox'.

Our work proposes another step in the direction of accurate boundary detection, by pushing further the machine learning approach. In this work we reconsider the observation made in earlier works e.g. [4,9], where it was mentioned that using more elaborate machine learning techniques does not significantly improve performance. As we demonstrate here, while using the same cues as [8] we obtain better results based on a combination of techniques developed around boosting.

For this we build on the Anyboost framework [11] that views Boosting as gradient descent in function space. Based on this more general point of view, we first develop a

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 650–663, 2010.

new variant of boosting that optimizes an approximation to the F-measure of our classifier during training, instead of the exponential loss which is commonly minimized by Boosting. As boundary detectors are evaluated based on their F-measure, it is natural to expect that training also with the F-measure as a cost function will improve performance. We note that this contribution can be of broader interest, as the F-measure is employed in several other problems, such as retrieval, to deal with the case where the negative class largely outnumbers the positive one.

Second, we deal with ambiguity in the labelling of points by treating the orientation of the boundary as a hidden variable, and train our classifier using Multiple-Instance Learning [12].

Third, we leverage information from the whole dataset during training, instead of using a small set of points, as is commonly the case in other works. As the whole set of feature-label pairs cannot fit in memory, we use a stochastic gradient descent method, inspired from the recent Filterboost work [1]. At each round of boosting a subset of 'interesting points' is chosen and used to construct the weak learner for that round. This is done in a proper way in the setting of Anyboost, by forming a stochastic approximation to the functional gradient of the training cost with respect to the classifer. We can thus train complex classifiers without fear of overfitting, thanks to the huge number of available training samples ($\simeq$ 200 Images x 150000 Pixels).

A further improvement in performance is provided by discriminative information extracted from appearance descriptors. As in recent works [13] we compute descriptors densely on the image, thereby capturing the context of any given point, and provide this as an input to a classification algorithm. This provides additional information, that complements the local features used in the Berkeley edge detector.

Our contributions are experimentally evaluated on the Berkeley Segmentation Benchmark, demonstrating systematic improvements over the current state-of-the-art. As we intend to provide the source code for our work, we omit several implementation details; we focus on the major new ideas, leaving a more detailed presentation of the low-level processing for a longer version of this work.

2 Previous Work

After decades of edge detection research driven by insight and guesswork, a quantum jump has been the introduction of ground-truth labeled datasets [3,4] and the phrasing of edge detection as a pattern recognition task. The 'Berkeley edge detector' [4] was shown to outperform most edge detection approaches developed in the previous decades, by replacing intuitively developed measures, such as the strength of directional derivatives [14,15], with statistical measures of texture, color and intensity discontinuity, and leaving their combination to machine learning.

This approach has led to improved detectors based on Boosting [6], topological properties of the image [7], spectral gradients [8], multiscale processing [9] and sparse dictionaries [10], among others. The most powerful boundary detectors currently in use [8,16,17] rely on combining different cues for boundary detection, such as texture gradients, brightness/color gradients, or information extracted from spectral clustering. Each of these cues is indicative of the presence of an edge, and the task of learning their

optimal linear combination is typically accomplished with logistic regression. A point mentioned repeatedly in several works, e.g. [9,4] is that using more intricate machine learning tools does not substantially improve performance. However both [6] and our work indicate that substantial improvements can be obtained by using more sophisticated learning approaches, as we will now present.

3 Learning Boundary Detection

We start with a presentation of boosting, where we introduce notation and the Anyboost technique [11] which is central to the rest of the paper.

Our training data come as sets of input-output pairs, $(X_i, y_i), X_i \in \mathcal{X}, y_i \in \mathcal{Y}, i = 1, \ldots, N$, where N is the size of our training set, typically $\mathcal{X} = R^d$ and for classification $Y = \{-1, 1\}$. Boosting algorithms learn a mapping from the input to the output space using a linear combination of simpler functions ('weak learners'):

$$f_T(X) = \sum_{t=1}^{T} a_t h_t(X). \tag{1}$$

The weak learners h_t are members of a family of simple functions, $\mathcal{H}$ but their combination in f_T can result in a complex classifier ('strong learner').

Boosting algorithms construct f in a sequential manner, by introducing at each iteration t a new component $h_t \in \mathcal{H}$ and a corresponding coefficient a_t that will most quickly improve the performance of the classifier. Performance is quantified by a cost $C(f)$ for the discrepancy between the classifier's predictions and the labels of the training set. This is typically a sum of individual costs over the training set, i.e.:

$$C(f) = \sum_{i=1}^{N} c(f(X_i), y_i) \tag{2}$$

For instance $c(f(X_i), y_i) = \exp(-y_i f(X_i))$ gives the exponential loss used in Adaboost training [18], while logistic regression scores have been considered in [19]. Moreover, different algorithms have been proposed to perform the selection steps for the weak learner and stepsize including Discrete-Adaboost [18], Gentleboost [20], or Confidence-Rated Boosting among others.

A unifying theme for these algorithms has been provided by the Anyboost algorithm [11], that views boosting as gradient descent in function space; namely each round of boosting can be seen as moving the function f in the direction that most rapidly decreases $C(f)$. In specific, consider that the outputs of the classifier f_t at iteration t on the training set are combined in a vector $\mathbf{f}$, s.t. $\mathbf{f}_i = f_t(X_i), i = 1, \ldots, N$. The negative gradient of the cost function with respect to the classifier's responses:

$$\mathbf{g}_i = -\frac{\partial C}{\partial \mathbf{f}_i}, \tag{3}$$

provides the update direction for f on the training set that will most rapidly decrease the cost being optimized. As we can only change our classifier by adding a member

of the family $\mathcal{H}$, boosting resorts to finding that function $h^* \in \mathcal{H}$ that is closest to the direction pointed by $\mathbf{g}$, i.e. has maximum inner product with $\mathbf{g}$:

$$h^* = \text{argmax}_h < \mathbf{g}, h >= \text{argmin}_h l(h), \quad l(h) = \sum_i \frac{\partial C}{\partial \mathbf{f}_i} h(X_i). \tag{4}$$

At each round we thus train a classifier using a reweighted version of the training set; each sample has a weight $|g_i|$, while the sign of g_i determines whether the weak learner should have a positive response. At each round a weak learner $h_t \in \mathcal{H}$ is chosen so as to minimize Eq. 4.

Once the h_t is chosen, its coefficient a_t is determined with line search to minimize $C(f_{t-1} + a_t h_t)$. The Anyboost algorithm is thus summarized as follows:

$f_T = \text{ANYBOOST}(C, \{X_i, Y_i\}, i = 1 \ldots N, T)$
Set $\mathbf{f}_i = 0, \forall i$
for $t = 1$ to T **do**
(a) Compute negative Gradient of C at $\mathbf{f}$: $\mathbf{g}_i = -\frac{\partial C}{\partial \mathbf{f}_i}$.
(b) Find the weak learner h_t which minimizes: $l(h) = \sum_i -\mathbf{g}_i h(X_i)$
(c) Choose the step size a_t that minimizes $C(f_{t-1} + a_t h_t)$ using line search.
(d) Set $\mathbf{f}_i = f_t(X_i)$.
end for
Output $f_T(x) = \sum_{t=1}^{T} a_t h_t(x)$.

We can now proceed with the presentation of our contributions in using Boosting for boundary detection. These are in the following directions:

- Using a cost C that properly measures the performance of our boundary detector system, by approximating its F-measure, Sec. 3.1.
- Dealing with ambiguity in labeling using Multiple-Instance Learning in conjunction with Anyboost, Sec. 3.2.
- Exploiting the whole Berkeley training set by forming a stochastic approximation to the weak-learner training criterion, l, Sec. 3.3 and the cost C used during line-search, Sec. 3.3.

3.1 F-Measure Boosting

Most training criteria in Boosting are defined as summations of a sample-based cost function over the whole training set, as in Eq. 2. This is the case for instance in the exponential loss or the log-likelihood score. However, such criteria can lead to poor classifiers when the training sets are imbalanced, which is the case for boundary detection: there are two orders of magnitude less boundary points than non-boundary points, so a classifier that errs in favor of non-boundary decisions can have a lower score than a more balanced one, when using a summation-based cost.

This is reflected in the F-measure that is used to score boundary detectors, defined as the geometric mean of the classifier's *precision*, p and *recall*, r:

$$F = \frac{2pr}{p+r}, \quad \text{where} \quad p = \frac{TP}{TP+FA}, \quad r = \frac{TP}{TP+MS} \tag{5}$$

In Eq. 5 TP is the number of the true positives, MS the number of misses, and FA the number of false alarms. Precision gives us the proportion of correct detector responses, while recall indicates the proportion of the true boundaries that have been detected. Note that the false negatives do not appear anywhere; therefore the classifier does not get credit for rejecting negatives, but only pays for false alarms. This allows to deal with a large negative class during evaluation (and training).

Even though the F-measure is broadly used as an evaluation measure, it is not commonly used in training as it is harder to optimize. However, algorithms for optimizing it have been developed in the context of SVMs [21] and logistic regression [22], while the authors in [8] mention optimizing the F-measure to training their logistic regression-based classifier. Here we use the Anyboost framework to apply the ideas developed in [22] to classifiers trained with boosting.

Following [22], we express the TP, MS, FA terms as sums over the training set:

$$TP=\sum_{i=1}^{N}[\hat{y}_i=1][y_i=1],\ FA=\sum_{i=1}^{N}[\hat{y}_i=1][y_i=-1],\ MS=\sum_{i=1}^{N}[\hat{y}_i=-1][y_i=1] \quad (6)$$

where $\hat{y}_i$ is the label estimated by the classifier (e.g. by thresholding $f(X_i)$ at 0), y_i is the correct label and $[\cdot]$ is indicating the truth of $\cdot$.

We then replace the quantities in Eq. 6 with probabilistic approximations, by replacing $[\hat{y}_i = 1], [\hat{y}_i = 0]$ with the soft measures $P(y = 1|f(X_i)), P(y = -1|f(X_i))$ respectively. For instance, we approximate the number of false alarms by $FA \simeq \hat{FA} = \sum_{i=1}^{N} P(y_i = 1|f(X_i))[y_i = -1]$. We combine the estimates of precision $\hat{p} = \frac{\hat{TP}}{\hat{TP}+\hat{FA}}$ and recall $\hat{r} = \frac{\hat{TP}}{\hat{TP}+\hat{MS}}$ in an approximation to the classifier's F-measure:

$$\hat{F} = \frac{2\hat{p}\hat{r}}{\hat{p}+\hat{r}} = \frac{\hat{TP}}{\hat{TP}+(\hat{FA}+\hat{MS})/2} \quad (7)$$

We thereby replace the terms showing up in the original F-measure by differentiable quantities, that smoothly vary as we change the classifier's output. This allows us to perform gradient descent so as to maximize the approximate F-measure. We now present how to optimize this measure with a classifier trained with Anyboost, using as cost $C(f) = 1 - \hat{F}$.

For now, we consider turning the output of a boosting-based classifier into a soft estimate by setting $P(y = l|f(X)) = \sigma_l(f(X)) = \frac{1}{1+\exp(-lf(X))}$, where l indicates the label of the training point. In Sec. 3.2 we will present a more elaborate expression, that can be directly incorporated in what we now present.

To apply the Anyboost algorithm, we need to measure how changing the response of the classifier at point i will affect the classifier's F-measure. This is given by [22]:

$$\mathbf{g}_i = \frac{\partial C}{\partial \mathbf{f}_i^t} = \left[H[y_i=1] - \frac{H^2}{2}\hat{TP}\right]\sigma'_{y_i}(\mathbf{f}_i^t),\ \ H = \left(\sum_{i=1}^{N}[y_i=1] + \frac{1}{2}(\hat{TP}+\hat{FA})\right)^{-1}, \quad (8)$$

while for a sigmoidal σ we have $\sigma'_{y_i}(\mathbf{f}_i) = \frac{d\sigma_{y_i}(\mathbf{f}_i)}{\mathbf{f}_i} = \sigma_{y_i}(\mathbf{f}_i)(1-\sigma_{y_i}(\mathbf{f}_i))$.

The expression in Eq. 8 determines the weighting of point i for round t, using the classifier f_{t-1} from the previous round: The vector $-\mathbf{g}$ indicates how the classifier's outputs should change so as to most rapidly increase the F-measure; and as already mentioned, with Anyboost we choose the $h_t \in \mathcal{H}$ that is closest to this direction, by maximizing $\sum_i -\mathbf{g}_i h_i(X_i)$.

We note that the cost function is *not* defined as a sum of individual costs, but rather is combining nonlinearly two global measures, the classifier's precision and recall. But at each round we compute the partial derivative of the cost w.r.t. the classifier's output on the individual points, which is forming a local linear approximation to the cost; this is then used to drive the fitting of the weak learner. Of course, the optimization cost is no longer convex so we may end up in local minima of the cost; nevertheless in our results we observed that the performance of the classifier trained with this criterion is better than that of the one trained with the convex criterion of standard Adaboost.

3.2 Multiple Instance Learning with Noisy-OR

So far we have been considering that we are provided with feature-label pairs. But our classifiers use orientation-dependent features, and classify each point based on an assumed orientation, say j; a point is labeled as positive if the classifier fires along *any* orientation. Using manual annotations to determine the orientation is tricky, since different users may suggest different orientations for the same image location depending on the granularity of their segmentation (e.g. on texture boundaries). Moreover, for points such as corners, or junctions, orientation is not properly defined.

To deal with this we use the adaptation of Multiple Instance Learning to Boosting by [12], and train a classifier in a way that copes with the missing orientation information. In specific, we extract features X for our classifier at all N orientations (we use $N = 8$), obtaining a 'bag' of features $\mathcal{X}_i = \{X_{i,1}, \ldots, X_{i,N}\}$ at each point i. For each orientation our classifier provides us with a probability estimate $P(y_i = 1|X_{i,j}) = 1/(1 + \exp(-f(X_{i,j})))$. The final decision is taken by a Noisy-OR combination:

$$p_i = P(y_i = 1|\mathcal{X}_i) = 1 - \prod_{j=1}^{N} (1 - P(y_i = 1|X_{i,j})). \tag{9}$$

This has a similar behavior with the a maximum-based combination - the left hand side is large when any of $P(y_i = 1|\phi_j)$ is large, and small only when all of them are small - but is differentiable. We use p_i as a shorthand for the result of the noisy-or combination rule.

This allows us to train this classifier using gradient descent and in specific, with Anyboost. We now refine our earlier presentation: the probabilistic estimates used in the approximation of the F-measure in Sec. 3 correspond to the left-hand side of Eq. 9, namely the combined decision about the point after considering all orientations. However, the classifier that we train shows up in the the right hand side, and gives the probability of point i being an edge using the features $X_{i,j}$ computed for orientation j: $P(y = 1|X_{i,j}) = (1 + \exp(-f(X_{i,j})))^{-1}$.

Using Anyboost we consider the classifier responses $f_{i,j}$ on all possible orientations, and stack the derivative of the cost with respect to them in a vector $\mathbf{n}$. With a slight abuse of notation we use two indexes for the elements of this vector and denote its elements by $\mathbf{n}_{i,j} = \frac{\partial C}{\partial f_{i,j}}$. The partial derivatives can be computed using the chain law:

$$\frac{\partial C}{\partial \mathbf{f}_{i,j}} = \frac{\partial C}{\partial \mathbf{p}_i} \frac{\partial \mathbf{p}_i}{\partial \mathbf{g}_{i,j}} = \left[H[y_i = 1] - \frac{H^2}{2} \hat{TP} \right] (1 - p_i) p_{i,j} y_i. \tag{10}$$

The bracket on the left comes from Eq. 8 while the right hand side can be derived using the property of the sigmoidal $\sigma' = \sigma(1 - \sigma)$.

The weak learner is thus trained by maximizing $\sum_{i,j} n_{i,j} h(X_{i,j})$. An inspection of Eq. 10 reveals that this assigns higher weights to the orientations which give higher responses, and can thus drive more quickly the change in the cost function.

3.3 Filtering via Stochastic Gradient Descent

Up to now we have considered that at each round a weak learner is trained by optimizing a quantity obtained by summing over the whole training set as dictated by the Anyboost algorithm; e.g. for noisy-or we considered minimizing $\sum_{i,j} n_{i,j} h(X_{i,j})$ where i ranges over all pixels in all images and j ranges over 8 possible orientations.

In practice this can be infeasible, due to both time and memory constraints. It is therefore common practice to pick at random a subset of the training set initially and then use it throughout training. This can lead however to overfitting, in particular if a small training set is given and a complex classifier is trained, while an unfortunate choice of a subset for training can also result in poor performance. We would like instead to maintain the whole training set throughout training, and use a proper portion of it at each round.

For this we propose a solution inspired from the recent Filterboost work [1], that adapts Boosting to the filtering problem; the filtering problem amounts to iteratively training a classifier with a subset of the training set at a time, while guaranteeing its good performance over the whole training set.

The adaptation of this idea is straightforward, once the Anyboost interpretation of Boosting is developed: we replace the criterion $l(h) = \sum_i \mathbf{g}_i h(X_i)$ used in Adaboost with a stochastic approximation, $\hat{l}(h)$ obtained by using a subset of the training data. In specific, we first normalize $\mathbf{g}$ so that $\sum_i |\mathbf{g}_i| = 1$. This does not affect the choice of h. We then construct a distribution $p_{\mathbf{g}}(i) = |\mathbf{g}_i|$ on the training set and interpret $l(h)$ as the expectation of $\mathrm{sgn}(\mathbf{g}_i) h(X_i)$ with respect to this distribution: $l(h) = E_{p_{\mathbf{g}}}(\mathrm{sgn}(\mathbf{g}) h(X))$.

We can then form a Monte Carlo approximation to this expectation, by drawing samples from the training set according to $p_{\mathbf{g}}$, and averaging the value of $\mathrm{sgn}(\mathbf{g}_i) H(X_i)$ on those samples:

$$l(h) = E_{p_{\mathbf{g}}}(\mathrm{sgn}(\mathbf{g}) h(X)) \simeq \frac{1}{K} \sum_{k=1}^{K} \mathrm{sgn}(\mathbf{g}_k) h(X_k) \equiv \hat{l}(h) \tag{11}$$

where $X_k, k = 1 \dots K$ are samples drawn from p_g. We thus replace the original problem of optimizing $l(h)$ by the optimization of its approximation $\hat{l}(h)$, formed using K instead of N samples. In practice, while our training set contains $N \simeq 3 \cdot 10^7$ points, we use $K = 5 \cdot 10^5$ samples at each iteration.

Using this scheme the points are chosen adaptively at each iteration: they are drawn from p_g, which is quantifying their usefulness for decreasing the cost *at the current round*. This allows our training algorithm to make the best use of the training data, and focus on the harder ones from the whole training set. Contrary, if one works with a fixed subset of the training set throughout, boosting fine tunes the performance of the classifier over a small set of points which leads to diminishing returns, or even overfitting for larger rounds of boosting.

We note that the proposed technique may seem similar to the stochastic Boosting method of [23]; however in our case the choice of points is driven by the cost gradient at the current round, instead of being a random sampling of the training set. Moreover, the same approach can be used to optimize any other training cost, and is not constrained to the F-measure used here. We therefore believe it can prove useful for a broader range of problems apart from feature detection.

Step size selection. The scheme described above allows us to find the approximately optimal weak learner h_t at round t; a similar scheme can be used to estimate the optimal step size a_t using a stochastic approximation to the cost function. As is known from Monte Carlo integration, forming a good stochastic approximation of a quantity requires sampling it more densely where it is larger in magnitude. We therefore use a different set of samples to estimate the step-size, by sampling more densely points that contribute to the $\hat{TP}, \hat{FA}, \hat{MS}$ quantities used to estimate the F-measure. For this, we form the function $d_i = y_i + p_i$ that adds the two quantities which indicate whether the response p_i at point i, can affect the F-measure: being a true positive/false negative, in which case $y_i = 1$ or being a false positive, in which case p_i is large. We normalize d_i so that it sums to one, and we see it as a distribution on lthe training set, denoted by p_d.

We then express $\hat{TP}, \hat{FA}, \hat{MS}$ as expectations with respect to this distribution, and form Monte Carlo approximations to these; for instance for $\hat{TP}$ we have:

$$\hat{TP} = \sum_{i=1}^{N} p(y=1|\mathcal{X}_i) = \sum_{i=1}^{N} d_i \frac{p(y=1|\mathcal{X}_i)}{d_{i,j}} = E_{p_d}\left(\frac{p(y=1|\mathcal{X}_i)}{d_{i,j}}\right), \quad (12)$$

so $\hat{TP} \simeq \sum_{k=1}^{K} p(y=1|\mathcal{X}_k)/d_k$ where X_k are samples drawn from p_d. A sample here amounts to the whole bag $\mathcal{X} = \{X_{i,1}, \dots, X_{i,N}\}$ at point i.

Summarizing, the optimal step a_t is found at each round t using line search, based on the stochastic approximation to $C(f_{t-1} + a_t h_t)$. When estimating the value of C for a candidate step a_t we perform the following steps:

- Compute the classifier's response $f(X_{i,j}) = f_{t-1}(X_{i,j}) + a_t h_t(X_{i,j})$ for all samples $i = 1 \dots K$ and all instances $j = 1, \dots 8$, within each bag.
- For each sample i, combine these responses using the noisy-or combination rule, to provide $p_i = p(y=1|\mathcal{X}_i)$.

– Form the Monte-Carlo approximations to $\hat{TP}, \hat{MS}, \hat{FA}$ as in Eq. 12 and combine them to provide an estimate $\hat{f}$.

4 Discriminative Features from Descriptors

In our earlier work [24] we have observed that a substantial improvement in performance can be gained by extracting discriminative information from descriptors. A similar result was preliminarily observed in [25] for the figure-ground assignment task, where geometric blur descriptors were used to leverage mid-level information.

Here we develop this idea further, and demonstrate the gain obtained by integrating descriptors with the other cues used during boosting. Specifically, in [24] we extract SIFT descriptors at multiple scales around candidate edgels to form a high-dimensional feature vector describing the context in which each edgel appears. We extend this idea by *densely* computing descriptors over the image; we can thus use their low-dimensional projections as regular features during both training and testing. Instead, in our earlier work we needed to do boundary detection and non-maximum suppression at the very beginning to extract descriptors around candidate edges. We thus replace our original two-tiered detection with an integrated version. More importantly, we experimentally demonstrate that even when using a highly optimized detector, using descriptor information yields an additional gain in performance.

We use a log-polar sampling scheme as in [13,26], using a sampling grid with 5 scales and 12 angles. Compared to SIFT, the log-polar sampling allows us to re-use the same descriptor for multiple orientations, simply by permuting its indexes, while also taking into account the context from a larger part of the image.

As in the Daisy descriptor [13], at each sampling point we compute derivative-of-Gaussians along eight orientations; the scales of the Gaussians are set proportional to scale of the point, and we compute such descriptors for all three channels of the Lab space. We also use Gabor filters for the L-component to capture texture information. Both Gabor and Gaussian filters are implemented using recursive (IIR) filtering to speedup descriptor computation. In all, we have 4 channels (3 for Lab and 1 for Gabor-texture), with 6 scales, 12 radii, and 6 orientations each, giving us a high-dimensional descriptor of the context around a point.

As in [24] we use a pre-processing step that discriminatively compresses descriptors into a low-dimensional space, and then use the coordinates in this space as inputs to our classifier. In specific, we use the Spliced Average Variance Estimation technique of [27] to find such a projection; this provides us with a set of orthogonal projection directions that can be easily computed in test-time.

In Fig. 1 we visualize the first two projection directions for descriptors extracted around a presumably horizontal edge. We show two different projections (vertical direction) for three cues (horizontal direction). Each projection is computed by summing the products of the descriptor values with the corresponding projection elements. Each needle shows the matrix entry for the corresponding location and orientation of the descriptor: red/blue denotes sign while the length indicates magnitude. Even though not as easily interpretable as the projections we would obtain from PCA, we observe that the projection dimensions correspond to geometrically meaningful patterns.

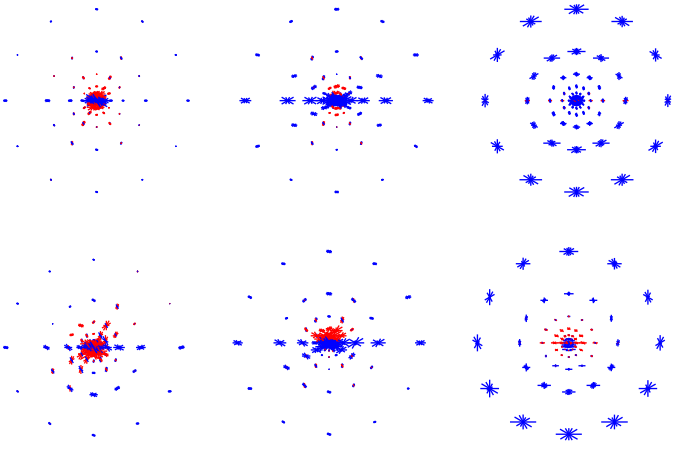


Fig. 1. Discriminative projections computed for (left) intensity (middle) color and (right) texture descriptors. The color of the needles indicates the sign, and their length indicates the magnitude of the projection coefficient for the corresponding descriptor dimension.

5 Application to Boundary Detection

We now focus on the problem of boundary detection. We first describe an adaptation of F-measure boosting that proved beneficial in tuning our detector, and then provide experimental results.

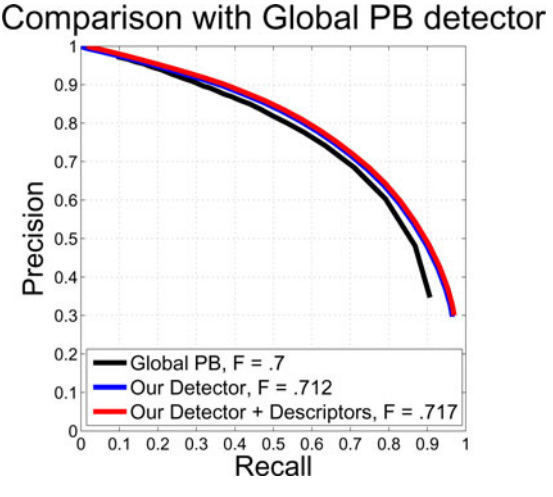


Fig. 2. Benchmarking results. Our method achieves an F-measure of 0.712, while together with descriptors the performance increases to 0.717. This compares favorably to the global-Pb detector, whose reported F-measure is .70. Please see text for details.

Fig. 3. Sample results from the Berkeley benchmark: the ground truth is shown on the middle and on the right we show our detector's estimate for the probability of a boundary

5.1 Calibrating the F-Measure for Boundary Detection

So far we have considered training a classifier with F-measure boosting in a general setting. For the boundary detection task in specific, we have realized that the reported F-measure in the evaluations is affected by two additional factors: First, images in the Berkeley benchmark are labeled by multiple persons, so certain points receive a 'boundary' label multiple times. We therefore take into account the number of times N_i that each training point i was labelled as positive in the expressions for TP and TM:

$$TP = \sum_{i=1}^{N} N_i[\hat{y}_i = 1] \simeq N_i P(y_i = 1|f(X_i)) = \hat{TP} \tag{13}$$

$$MS = \sum_{i=1}^{N} N_i[\hat{y}_i = -1] \simeq \sum_{i=1}^{N} N_i P(y_i = -1|f(X_i)) = \hat{MS} \tag{14}$$

The expressions for TP and MS are the ones that are computed during the evaluation, according to the code of [4], while the expressions for $\hat{TP}$ and $\hat{MS}$ are used for training. These expressions emphasize points that are consistently labeled by more users as boundaries. This improves the F-measure of the detector on both the training and test sets.

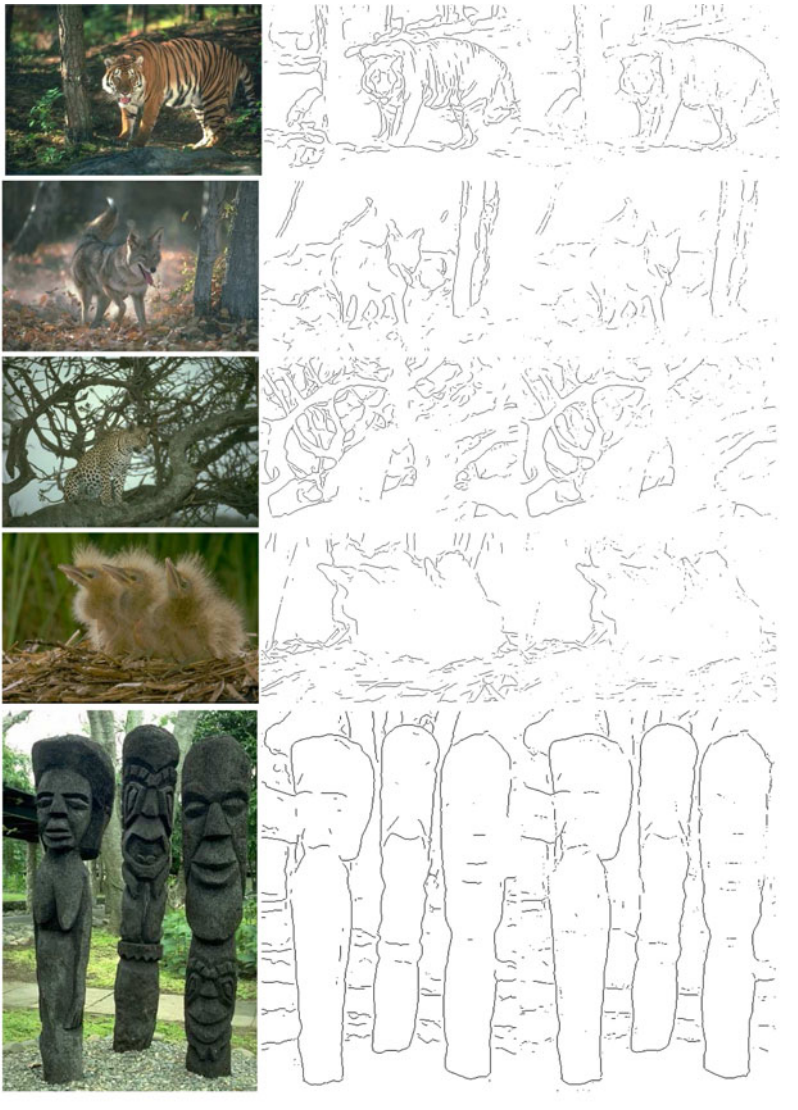


Fig. 4. Comparisons of the Global PB detector (left) with our results (right). Both detectors are thresholded at the value giving the best global F-measure. Overall, we observer that our detector responds less to textured, or cluttered image areas.

Second, the boundaries used for evaluating the detector are obtained after non-maximum suppression. Only a fraction of false positives will thus survive suppression, and give a false alarm. Scaling the estimate of $\hat{FA}$ in Eq. 14 by a fraction of $1/10$ therefore yields an estimate of false alarms that is much closer to the one reported by the evaluation software.

5.2 Experimental Results

In order to systematically evaluate our approach we have conducted systematic experiments on the Berkeley Benchmark.

As mentioned in the introduction, our contributions are in both the learning, and the feature extraction direction. To validate our contribution in learning, we first train a classifier using exactly the same features as in [8], namely multi-scale color and texture gradients, as well as 'spectral gradients', obtained from the directional derivatives of the eigenvectors found from normalized cuts. The difference is in the learning algorithm (Adaboost) and the fact that we use the whole training set for training. Both we and [8] use the F-measure for training, so we are optimizing essentially the same cost. In [17] a combination of the gPb detector with a segmentation algorithm results in a improvement of the gPb detector's F-measure from .7 to .71. Our detector achieves an F-measure of .712 while not using additional information from segmentation. The performance of the classifier trained using our earlier setup [24] of using a sparse set of training data, with fixed orientation decreases the performance to $F = .7$. This demonstrates the merit of using MIL for orientations and Filterboost.

To validate our contribution in feature extraction we perform the training procedure, but now introducing the new features obtained from the appearance descriptors by discriminative dimensionality reduction. It becomes clear that the descriptors provide an additional boost in performance, which increases to .717.

Regarding testing time, extracting the features of [8] takes 80s on a 3Gh machine, while [16] cut it down to 1s on a GPU. Computing dense descriptors requires 50s in Matlab, but is also easily parallelizable on GPUs. Once features are extracted, evaluating our detector takes 20s in Matlab.

6 Conclusions

In this work we have pushed further the machine learning approach to one of the most basic problems in computer vision, boundary detection. We have obtained state-of-the-art results in the setting of boosting by (i) using a proper training criterion, based on the F-measure (b) exploiting the whole training set during training and (iii) introducing new discriminative features using context information captured by descriptors. In future work we intend to extend the application of these ideas to the detection of other types of low-level features, while also pursuing the exploitation of these better boundaries for object recognition.

References

1. Bradley, J.K., Schapire, R.E.: Filterboost: Regression and classification on large datasets. In: NIPS (2007)
2. Marr, D.: Vision. W.H. Freeman, New York (1982)

3. Konishi, S., Yuille, A.L., Coughlan, J.M., Zhu, S.C.: Statistical Edge Detection: Learning and Evaluating Edge Cues. PAMI 25 (2003)
4. Martin, D., Fowlkes, C., Malik, J.: Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues. PAMI 26, 530–549 (2004)
5. Ren, X., Fowlkes, C., Malik, J.: Scale-invariant contour completion using crfs. In: ICCV (2005)
6. Dollar, P., Tu, Z., Belongie, S.: Supervised Learning of Edges and Object Boundaries. In: CVPR (2006)
7. Arbelaez, P.: Boundary Extraction in Natural Images Using Ultrametric Contour Maps. In: WPOCV (2006)
8. Maire, M., Arbelaez, P., Fowlkes, C., Malik., J.: Using Contours to Detect and Localize Junctions in Natural Images. In: CVPR (2008)
9. Ren, X.: Multiscale helps boundary detection. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 533–545. Springer, Heidelberg (2008)
10. Mairal, J., Leordeanu, M., Bach, F., Hebert, M., Ponce, J.: Discriminative sparse image models for class-specific edge detection and image interpretation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 43–56. Springer, Heidelberg (2008)
11. Mason, L., Baxter, J., Bartlett, P., Frean, M.: Boosting algorithms as gradient descent. In: NIPS (2000)
12. Viola, P., Platt, J.C., Zhang, C.: Multiple Instance Boosting and Object Detection. In: NIPS (2006)
13. Tola, E., Lepetit, V., Fua, P.: A fast local descriptor for dense matching. In: CVPR (2008)
14. Canny, J.: A Computational Approach to Edge Detection. PAMI 8, 679–698 (1986)
15. Perona, P., Malik, J.: Detecting and Localizing Edges Composed of Steps, Peaks and Roofs. In: ICCV, pp. 52–57 (1990)
16. Catanzaro, B., Sundaram, N., Su, B., Lee, Y., Murphy, M., Keutzer, K.: Efficient high-quality image contour detection. In: ICCV (2009)
17. Arbelaez, P., Maire, M., Fowlkes, C., Malik., J.: From contours to regions: An empirical evaluation. In: CVPR (2009)
18. Freund, Y., Schapire, R.: Experiments with a new Boosting Algorithm. In: ICML (1996)
19. Collins, M., Schapire, R.E., Singer, Y.: Logistic regression, adaboost and bregman distances. In: Machine Learning (2000)
20. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. Ann. Stat. (2000)
21. Joachims, T.: A support vector method for multivariate performance measures. In: ICML (2005)
22. Jansche, M.: Maximum expected f-measure training of logistic regression models. In: HLT 2005 (2005)
23. Friedman, J.H.: Stochastic gradient boosting. Comput. Stat. Data Anal. (2002)
24. Kokkinos, I.: Highly accurate boundary detection and grouping. In: CVPR (2010)
25. Ren, X., Fowlkes, C.C., Malik, J.: Figure/ground assignment in natural images. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 614–627. Springer, Heidelberg (2006)
26. Kokkinos, I., Yuille, A.: Scale Invariance without Scale Selection. In: CVPR (2008)
27. Cook, D., Lee, H.: Dimension reduction in binary response regression. JASA 94 (1999)

Unsupervised Learning of Functional Categories in Video Scenes*

Matthew W. Turek, Anthony Hoogs, and Roderic Collins

Kitware, Inc., Clifton Park, N.Y. U.S.A.
{matt.turek,anthony.hoogs,roddy.collins}@kitware.com
http://www.kitware.com

Abstract. Existing methods for video scene analysis are primarily concerned with learning motion patterns or models for anomaly detection. We present a novel form of video scene analysis where scene element categories such as roads, parking areas, sidewalks and entrances, can be segmented and categorized based on the behaviors of moving objects in and around them. We view the problem from the perspective of categorical object recognition, and present an approach for unsupervised learning of *functional* scene element categories. Our approach identifies functional regions with similar behaviors in the same scene and/or across scenes, by clustering histograms based on a trajectory-level, behavioral codebook. Experiments are conducted on two outdoor webcam video scenes with low frame rates and poor quality. Unsupervised classification results are presented for each scene independently, and also jointly where models learned on one scene are applied to the other.

Keywords: functional modeling, unsupervised learning, video analysis.

1 Introduction

We present a new approach to video scene modeling and recognition that is based on unsupervised, location-independent segmentation of scene element categories. Existing work in video scene modeling has largely focused on segmenting dominant motion patterns [1,2,3,4,5] and significant regions such as track sources and sinks [1,6], given observed trajectories and detection algorithms for each scene element type. Here, we view the problem from the perspective of categorical object recognition, and present an approach for unsupervised learning and modeling of *functional* scene element categories – entities that are defined primarily by their behavior and/or surrounding activity, rather than their appearance or shape. Typical functional scene elements include vehicular traffic lanes, sidewalks, parking spaces, crosswalks, building entrances/exits, benches, bus stops, and so on. Many of these functional scene elements can not be distinguished

* This material is based upon work supported by the Defense Advanced Research Projects Agency under prime contract HR0011-06-C-0069, subcontract 070861. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 664–677, 2010.

by appearance alone. For instance, parking spaces and traffic lanes often appear nearly identical. However, for applications like normalcy modeling and abnormal event detection, it may be important to to understand the function of a region. Functional scene elements can also help identify the functional behavior of moving objects [16]. Identification of functional scene regions is important, then, as an enabling technology for event detection and normalcy modeling.

Inspired by the bag-of-words paradigm for object recognition, our approach identifies regions with similar behaviors in the same scene and/or across scenes, by clustering histograms based on a trajectory-level, behavioral codebook. The regions do not need to be spatially contiguous; rather, we seek regions that are spatially disjoint but have similar functional properties. A cluster of such regions (in feature space, not scene coordinates) corresponds to a functional category that can be assigned a conceptual label.

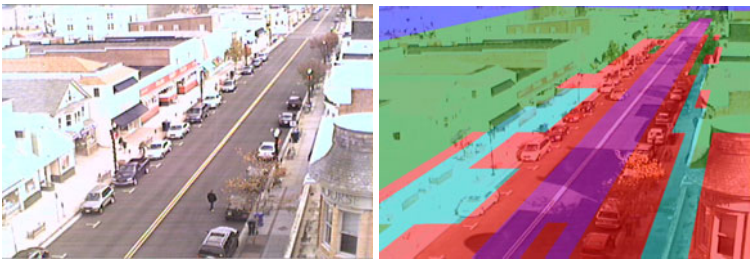


Fig. 1. Left: A typical scene from an outdoor webcam (Ocean City, NJ). **Right:** The result of unsupervised scene classification. Regions with the same color are determined to have the same functional type.

Functional object recognition was pioneered some time ago with work on recognizing chairs in static images [7]. Recently, work has appeared on integrating observed function in video with appearance for object recognition [8,9]. Maintaining distributions of track-level information at scene locations has been studied previously [3,10]. Codebooks of local kinematic and object features have been used to model motion patterns and detect unusual activities [4]. Swears and Hoogs [15] presented methods to detect specific scene elements which relied on hand-crafted detectors. Our work differs in that we do not attempt to segment the various motion patterns in a scene from each other, or to develop detection algorithms specific to any scene element category; instead we learn and classify functional regions. In addition, we learn models that are transportable across scenes.

Another recent technique is [11] where optical flow based features are combined with multi-scale analysis to learn motion patterns. The reliance on optical flow patterns adds robustness to tracking. Our motivating datasets are webcams whose low-frame rates (approx. 1-2 Hz.) provide large displacements with very sparse temporal sampling for the computation of optical flow features. Instead,

we incorporate tracking robustness by using a hierarchy of features, including detection, track, track fragment, and multi-track information.

At a high level, our method consists of: 1) developing a common, hierarchical feature-space representation for all behavioral scene element categories; 2) unsupervised learning of behavioral category models which are independent of scene location and transportable across scenes; 3) segmenting video scenes into the functional categories. To our knowledge, this has not been done previously. Our method can represent a variety of behavior-based scene elements in urban, outdoor scenes. The techniques can be used to learn a generic set of functional element categories across a variety of scenes, or to learn the specific element types present in one scene.

As mentioned previously, we operate on webcams, which are highly challenging due to poor quality and very low frame rates. An example result is shown in Figure 1, computed from 8 hours of poor quality, 1Hz video from a webcam. The camera was (manually) calibrated, and the scene was partitioned into a regular grid in the ground plane. Because of the low frame rate, banded noise and compression artifacts, detection and tracking were particularly noisy and error-prone. Nevertheless, grid cells with similar behaviors were successfully clustered to yield functional categories such as vehicular traffic lane, pedestrian traffic lane (sidewalk), parking spot, and building entrance.

The method is applicable to generic surveillance cameras as well as webcams. The latter introduces significant challenges because of poor tracking, but these are addressed through statistical methods and the hierarchical feature set as described below. Calibration to a ground plane is useful if not required; automatic calibration in video has been studied and is beyond the scope of this work, but will be utilized in future work.

The method has a few key limitations. Currently, it is assumed that a single grid cell or region is indicative of its functional type. This effectively bounds the minimum size of functional regions (spatial resolution of the grid), since a small portion of a scene element may have insufficient information. Similarly, cell neighborhoods are not directly considered during clustering, although some of the features weakly associate nearby cells. Another drawback is that temporal state transitions are not modeled explicitly. Scene elements with multiple behaviors at different times, such as intersections, are represented multiple modes in the codebook histogram. These shortcomings will be addressed in future work.

Our approach is described in Section 2. Section 3 presents our results and we conclude in Section 4.

2 Modeling Approach

Our approach adapts the bag-of-words concept to trajectory-level behavioral analysis. An overview of the method is shown in Figure 2. First, on each video scene, tracks are computed for a period of time that is sufficient to capture the range of activity in the scene (typically a few hours or a day depending on event density and scene complexity). We assume that the cameras are roughly

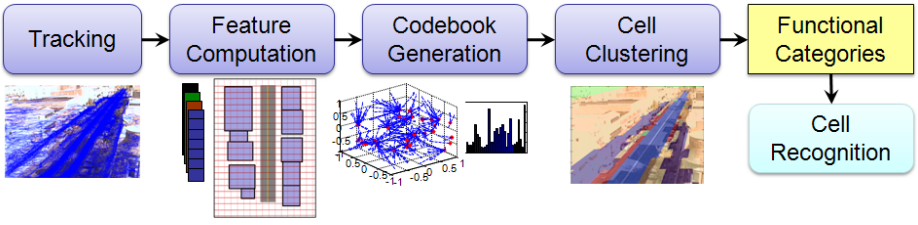


Fig. 2. The overall approach. Features describing local behaviors are computed for each track interval through each cell, and used to generate a codebook. Each cell is described by a codebook histogram. The histograms are clustered to form functional element categories, which are used for recognition.

calibrated to the ground plane, so that ground-plane tracking and normalization may be performed. Each video scene is partitioned into a set of regions, such as a regular spatial grid in the ground plane.

Next, a descriptive set of *behavioral features* is computed. For each track, and for each grid cell that the track intersects, detection and track-level features capture single-object, local, behavioral and object characteristics. Within each cell, the these features are accumulated to form feature distributions for the cell. In addition, cell-level features capture the relationship between cells traversed by the same track, and localized relationships between tracks over time. In total, the feature set characterizes local behaviors in the same way that patch descriptors characterize local appearance for object recognition.

For the scene shown in Figure 1, tens of thousands of feature vectors are computed from 8 hours of video. These feature vectors are clustered using a method such as mean-shift or K-means to form a codebook of a size that is comparable to the number of cells. For each cell, a codebook histogram is formed by finding the closest centroid for each feature vector in the cell.

For each scene, the cell histograms are clustered using mean-shift [12] on the L_2 histogram distance. Each cluster corresponds to a set of cells with a common local behavior pattern, i.e. the same functional category.

At this point each scene is segmented into functional types by cluster index, but the types are not textually labeled. This labeling is simply the assignment of a string name to each cluster index, which is done manually with little effort.

On a new video scene, the clusters can be used to perform recognition of functional elements. As in learning, the new scene is spatially partitioned and tracked, and codebook histograms are computed for each cell. Each cell is then recognized by initializing mean-shift with the cell histogram, and outputting the cluster found by mean-shift.

The following subsections describe these steps in more detail.

2.1 Moving Object Detection and Tracking

We used a standard background subtraction algorithm [13] to detect moving objects and then used two simple data association trackers (one tuned for pedestrians

and the other for vehicles) to link these objects into tracks. The data association-based tracking is similar to [14]. These algorithms are completely automatic. A rough projective camera is computed by hand, once per scene. This projective camera is used in the tracking algorithms to help estimate object ground-plane position, size and velocity.

2.2 Behavioral Feature Set Computation

We have developed a hierarchical feature set to provide robustness to tracking difficulties and to capture multiple levels of behavior detail. Our features include detection-based features, track-level features, and cell-level features.

Table 1. Detection level features, $\boldsymbol{d}_{j,k}$, for track j in cell k; track level features, $\boldsymbol{f}_{j,k}$ for for track j in cell k; and cell level features for cell k

	Element	Feature
$\boldsymbol{d}_{j,k}$	0	Median speed
	1	Median change in speed
	2	Median change in angle
	3	Median size
	4	Median detection aspect ratio
$\boldsymbol{f}_{j,k}$	0	Track length
$\boldsymbol{c}_k$	0	Number of track starts in cell
	1	Number of track stops in cell
	2	Number of tracks through cell
	3	Entropy of outgoing heading distribution
	4	Entropy of incoming heading distribution

Our detection, track-level, and cell-level features are listed in Table 1. The first block in the table contains the detection-based features, the second block contains the track-level features, and the third block contains cell-level features. Detection-based features incorporate information based solely on moving object detections that are within a particular cell. All the detections for a particular track are combined into one feature vector, using summary statistics. Typically, we use the median of the feature for the track samples (corresponding to a particular track) within a cell. Track level features $\boldsymbol{f}_{j,k}$ are computed for each track passing through a cell. Finally, cell level features $\boldsymbol{c}_k$ are computed across all tracks that pass through a cell.

The entropy of the heading distribution for incoming (outgoing) tracks is computed as in Equation 1, with the distribution taken over the heading of all start (stop) detections in a cell. Note that the entropy measure is independent of particular orientations, thus allowing this feature to be built into a codebook on one scene and then applied on a different scene.

$$E(\tau) = -\sum_i p(\theta_i)\ log\ p(\theta_i) \tag{1}$$

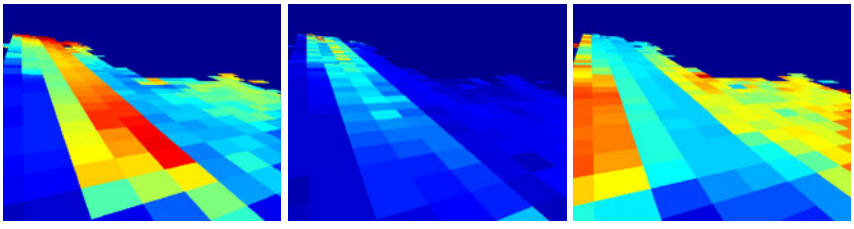


Fig. 3. Example features for the Ware scene. Size (left), speed (middle), aspect ratio (right).

where $p(\theta_i)$ is the probability of the heading of the i^{th} detection on track τ. We use a histogram of the headings along a track to model the heading probability.

It is important to choose features judiciously, as additional, non-informative, features can clutter the feature space and effectively add a noise term to feature distances which leads to misclassifications.

We create an ensemble feature vector $\boldsymbol{x}_{j,k}$ for each track j and cell k as:

$$\boldsymbol{x}_{j,k} = \begin{pmatrix} \boldsymbol{d}_{j,k} \\ \boldsymbol{f}_{j,k} \\ w_c \boldsymbol{c}_k \end{pmatrix} \tag{2}$$

where w_c is a weighting for cell features $\boldsymbol{c}_k$. The collection of all feature vectors $\mathcal{F}$ in a video sequence is:

$$\mathcal{F} = \{\boldsymbol{x}_{0,0}, \ldots, \boldsymbol{x}_{j,k}, \ldots, \boldsymbol{x}_{n,m}\} \tag{3}$$

for n tracks and m cells. One issue arises in simultaneously handling the detection, track, and cell-level features: there are more track level feature instances (one per track crossing a cell) than cell level features (one per cell). We handle this discrepancy by downweighting the influence of the cell features (through w_c). This allows us to build a single code book on a feature space including both track and cell feature dimensions. Currently we set $w_c = 1/n_{celltracks}$ where $n_{celltracks}$ is the number of tracks in a cell.

There are a few alternatives to the approach we have taken for combining the hierarchy of features. One alternative is to summarize all the track features into one feature vector per cell. This would inevitably lead to a reduction in information and cells with a multi-modal distribution of features would end up with a blended feature vector. On the detection level, we have summarized all the detections for one track in one cell into a single feature vector. Since the cells are relatively small and there are typically few detections for one track within a cell (< 3 is typical), this summarization is less problematic. However, across perhaps hundreds of tracks that pass through a cell, summarization of track level features would discard significant information. Another alternative solution would be to maintain two feature spaces, and create a codebook for each. One significant disadvantage of this approach, however, is that the centroids (Sec. 2.3) cannot capture joint information between the spaces.

2.3 Feature Codebook Generation

The use of codebooks or "visual words" has become immensely popular in visual recognition tasks, because they are an effective way of compactly representing high-dimensional, complex feature spaces. We apply them here because the behavioral feature space can be complex, but should also contain high-density areas corresponding to common activities. Stauffer and Grimson previously applied a codebook to activity analysis [4], where position was included and tracks were represented by sets of centroid labels. Here, we explicitly avoid dependence on absolute features such as position and heading, as our goal is to learn behavioral categories independent of locations or location-specific behaviors. Also, our codebook histogram is defined on cells, not tracks, which enables our hierarchical feature set.

All ensemble feature vectors in the video scenes, denoted $\mathcal{F}$, are used to create the codebook. Before clustering, each feature is normalized by its standard deviation across the observed data. We generate the codebook by clustering the ensemble features. We have experimented with both K-means and mean-shift and have found mean-shift to be considerably more stable than K-means with random initialization. Mean shift also has the advantage that K does not need to be specified, although the bandwidth parameter does impact performance. The mean shift for feature $\boldsymbol{x}_i$ with bandwidth parameter h and kernel $g()$ is defined as [12]:

$$\boldsymbol{m}_{h,G}(\boldsymbol{x}) = \frac{\sum_{i=1}^{n} \boldsymbol{x}_i g\left(\left\|\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right\|^2\right)}. \tag{4}$$

We typically use two iterations of mean shift on the set of features $\mathcal{F}$ to generate representative features. An Epanechnikov kernel is used to represent the underlying distribution, yielding a mean-shift implementation with a uniform kernel.

2.4 Cell Histogram Representation and Clustering

Once the codebook is defined, a codebook histogram is created for each cell. For each integrated feature vector in a cell, the closest codebook centroid is identified, and the corresponding bin in the histogram is incremented. To assure statistical significance, we ignore cells with too few feature vectors (< 20).

The final segmentation step is clustering cells that have similar histograms. We use mean-shift [12] for clustering, as we do not want to specify the number of clusters a priori, and we expect the clusters to be non-Gaussian and generally noisy. Spatial position of the cells is not considered, so that we can group cells with similar behaviors located in different parts of the same scene or in different scenes altogether. As mentioned above, we currently do not use cell neighborhood information in the clustering process, although some features are computed across neighboring cells.

Ideally, each of the resulting clusters would correspond to a single functional category such as road or parking spot, and each functional category would have

exactly one cluster. In practice, any of the categories may have a multi-modal feature distribution, in which case mean-shift should create multiple modes for one category. For recognition, all significant modes should be assigned the same label.

Generally, we expect that many scene elements will have non-trivial distributions of behavioral features. For simple categories, such as vehicular or pedestrian traffic lanes, it is strightforward to compute low-level features on individual tracks, and perform clustering on the raw features. For these cases, our use of cell-level features and codebook histogram clustering is perhaps more powerful (and more complex) than required.

Many other scene categories, however, have more complex and variable behaviors. Building entrances and exits, parking spots and crosswalks, for example, may exhibit a variety of behaviors even at the same locations. Our representation can support this, as long as the mode patterns are similar across the class. For one activity pattern in a multi-modal cell, the tracks in that pattern will give rise to feature vectors in histogram bin h_i. For a second activity pattern in the same cell, the tracks in that pattern will give rise to feature vectors in histogram bin h_j. The cell histogram will have two modes, and should be clustered with other cells that have the same two modes. This situation should arise for a cell outside a doorway, for example, if some people walk straight past the doorway (bin h_i) and others enter (bin h_j).

2.5 Scene Element Recognition

Once the clusters are formed, scene element recognition can be performed on a new video scene, or new parts of an existing scene. The spatial scale and partitioning of the new scene should be comparable to those used in training, and the scene should have comparable behaviors for the same scene categories (this may not be true across different regions of the world).

Each cell is recognized independently. Cell features are computed from its tracks as in training, and the cell histogram is created using the prior codebook. Each cluster is a collection of codebook histograms, and we label cells using mean-shift on the cell histogram, and outputting the cluster found by mean-shift.

3 Results

Experiments were conducted on many hours of video data, from two public webcams, with over 2500 tracks in each scene. Scene element learning, segmentation and recognition results are shown on each scene independently, and between scenes by learning on one and testing on the other.

3.1 Data

One webcam is in Ocean City, NJ, shown in Figure 1. Approximately 8 hours of data from one day was used. The frame rate is about 1-2 Hz, and the image

Fig. 4. The size of people in the Ocean City video. In the near-field, people are about 30 pixels in height; in the mid-field, about 16 pixels, and in the far-field, about 10 pixels. Compression artifacts are noticeable, particularly in the far-field.

size is 704×480. To provide a sense of scale and image quality, Figure 4 shows crops of a person in the scene.

The second scene is in Ware, UK. The frame rate is also ≈ 2 Hz, and the image quality is somewhat better than Ocean City (OC). The near-field has higher resolution, as the camera is mounted closer to the ground. The far-field resolution is comparable to OC. Shown in Figure 5, the scenes have a number of functional entities in common.

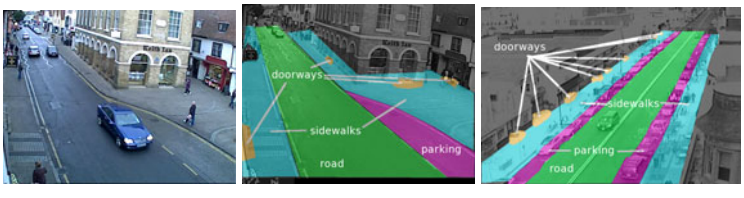


Fig. 5. The video scenes used in the experiments. Left is the Ware scene. Manually-generated ground truth labeling for Ware (resp. Ocean City) is the middle (resp. right). Ground-truth labeling is used for evaluation of results only.

To evaluate the algorithm, we manually created ground-truth labels for the roads, sidewalks, parking areas, and building entrances/exits as shown in the figure. These labels were not used by the algorithm in any way; they were used only for evaluation of the results.

3.2 Tracking

We used a background subtraction algorithm [13] to detect moving objects and a simple data association tracker to link these objects into tracks. The algorithms are completely automatic. We did not use any scene specific information to improve the moving object detection or tracking, except to use an approximate projective camera to compensate for the change in object size from the far-field to the near field.

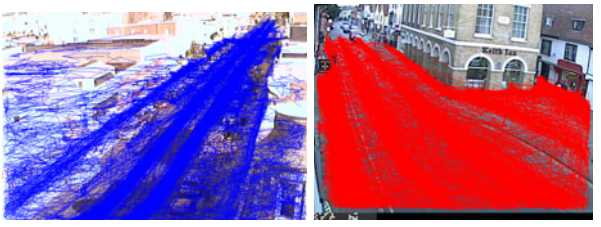


Fig. 6. Computed tracks on the Ocean City (left) and Ware (right) scenes

Figure 6 shows the computed tracks used in the experiments for both scenes. There are more than 2500 tracks in each scene, and the dominant motion patterns are clearly evident. Recall, however, that we do not attempt to segment these motion patterns, but rather to group all of the cells into a set of functional categories. Hence our desired result would match the ground truth in Figure 5, which does not separate the two lanes of the road.

While the tracking results are good overall, there is significant track fragmentation as pedestrians disappear under overhangs, signs and trees, or seem to disappear because of saturation effects. There are also a number of false tracks and track switches caused by false alarms in moving object detections. These are particularly evident near times of global lighting changes, because the false alarm rate rises until the background model catches up.

Our approach is quite robust against tracking errors because of its statistical nature. Many tracks are considered at each cell (≥ 20), and more video can be added as necessary.

3.3 Unsupervised Scene Segmentation and Classification

We conducted an initial experiment to evaluate the need for hierarchical features. We used the detection level features, denoted $\boldsymbol{d}_{i,j}$ in Table 1 as the only feature set, and proceeded to run the remainder of the algorithm, including code-book generation and cell clustering. Two representative results for the Ware scene are provided in Figure 7. (The ground truth for Ware is the middle image in Figure 5.) The detection level features are able to discriminate pedestrian/vehicle areas from the background. However, there is little ability to discriminate between the pedestrian areas and the roadway. In addition, the segmentation results are quite noisy.

To characterize the performance of the full system, including the full, hierarchical feature set, we conducted further experiments. We began by running experiments on the two web-cam datasets. In each case, a codebook was built on the scene and then that codebook was used to classify the cells in the scene.

On each scene, the system produces cluster IDs corresponding to scene element types, which are then scored against the ground-truth. In order to do this scoring, semantic labels must be assigned to the clusters. There is no clear-cut method

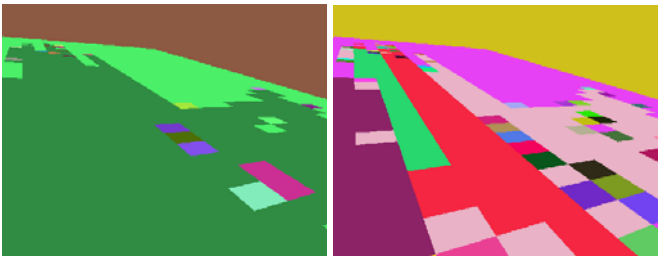
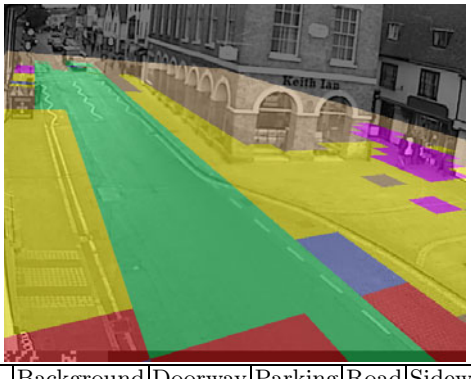


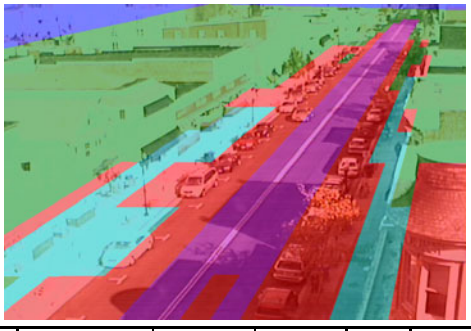
Fig. 7. Two representative clustering results on the Ware scene using detection features only. Several clustering experiments were performed using detection level features only, with the goal of separating pedestrian and vehicle areas. Results were either overly smooth (left) or quite noisy (right). While there is some discrimination of pedestrian/vehicle areas, the performance does not approach that of the proposed hierarchical features.



class	Background	Doorway	Parking	Road	Sidewalk	PCC
Background	391	11	1	1	5	0.9560
Doorway	0	4	0	0	4	0.5000
Parking	1	0	2	0	6	0.2222
Road	37	0	0	125	6	0.7440
Sidewalk	119	13	1	0	107	0.4458

Fig. 8. Top: Unsupervised classification result on Ware. Cells in the same cluster have the same color. Areas outside the ground-plane grid had too few tracks and were not considered. Other cells with fewer than the minimum number of tracks are blue and were not considered. **Bottom:** Confusion matrix. Each row represents the correct class, and each column is the computed class. The rightmost column is the per-class probabability of correct classification.

for this, as there may be multiple clusters that should correspond to one ground-truth label. Conversely, there may be clusters that encompass multiple labels. We assigned the semantic labels by hand, picking the label which visually made the most sense for the supplied clusters.



class	Background	Doorway	Parking	Road	Sidewalk	PCC
Background	621	0	20	19	7	0.9310
Doorway	2	0	0	0	2	0
Parking	1	0	31	0	1	0.9394
Road	0	0	11	39	0	0.7800
Sidewalk	7	0	20	0	15	0.3571

Fig. 9. Top: Unsupervised classification result on OC. **Bottom:** Confusion matrix.

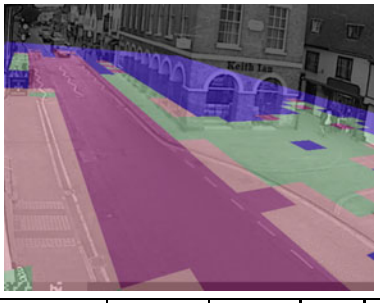
Results on the Ware and OC scenes (computed independently) are shown in Figures 8 and 9. All parameters are the same for both scenes, except that the bandwidth parameter for final clustering was adjusted to compensate for the different number of grid cells (this could be automated). The grid cell size is approximately the same as well. The codebook size is 80, computed with mean-shift. Each feature vector has dimension 11. The final clustering was also computed with mean-shift.

The results indicate that the more basic categories, road and sidewalk, are reasonably learned and classified. Doorways and parking are more difficult categories. There is some confusion between parking and sidewalk in both scenes, because people exit their cars and walk through the parking areas. The small parking area in Ware is partitioned into two clusters, because cars frequently drive through the blue cluster but rarely park there.

Doorways are detected in both scenes, but only partially. The doorway area on the far right in Ware is detected as cluster. There are three doors there, as well as a busy pedestrian thoroughfare just in front of the doors. A small patch in the upper left is clustered into the same class, because this area is a street corner where people emerge from a side street, and also pause while waiting to cross the main road.

3.4 Unsupervised Classification across Scenes

We also wished to understand the transferability of a codebook learned from features on one scene to a classification problem on another scene. In the next experiment, we learn scene element models (codebook and clusters) on one scene, still unsupervised, and "classify" cells in the other. Each cell is classified using



class	Background	Doorway	Parking	Road	Sidewalk	PCC
Background	391	0	0	4	14	0.9560
Doorway	0	0	0	1	7	0
Parking	1	0	0	1	7	0
Road	37	0	0	126	5	0.75
Sidewalk	119	0	0	7	114	0.4750

Fig. 10. Top: Unsupervised classification on Ware using models learned on OC. **Middle:** Classification scored against ground-truth for each cell. **Bottom:** confusion matrix.

mean-shift as described in the previous section. Semantic labels are applied with the same mapping from cluster index to semantic label used on the learning scene. The evaluation procedure is the same as for single-scene scoring. Results are shown in Figures 10 for models learned on OC and tested on Ware.

The results are quite comparable to those computed independently on Ware, indicating the that the method can generalize effectively beyond a single scene. The scenes are geometrically similar, but they are in different parts of the world, with different types of vehicles, buildings and so on. The cameras are different too, with slightly different frame rates, resolutions and quality. Traffic and pedestrian density is considerably higher in Ware.

Although not shown, we conducted the same experiment but with learning on Ware and testing on OC. The scores were slightly lower, but still comparable to the single-scene OC results.

4 Conclusion

We have developed a method that performs unsupervised classification of functional scene element categories observed in video. By abstracting away from specific locations and scenes, and by introducing a descriptive feature vector that characterizes local behavior, we learn generic behavior patterns that correspond to functional categories. Multiple, spatially-disjoint instances of the same scene element type can be identified within a scene, or between different scenes. Results are shown on two scenes and four categories – a small set, but the results are encouraging. In future work we plan to address the limitations outlined in the introduction.

References

1. Wang, X., Ma, K.T., Ng, G.W., Grimson, W.E.L.: Trajectory analysis and semantic region modeling using a nonparametric bayesian model (pdf). In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
2. Makris, D., Ellis, T.: Learning semantic scene models from observing activity in visual surveillance. IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, 397–408 (2005)
3. Hu, W., Xiao, X., Fu, Z., Xie, D., Tan, T., Maybank, S.: A system for learning statistical motion patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 1450–1464 (2006)
4. Stauffer, C., Grimson, E.: Learning patterns of activity using real-Time tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 747–757 (2000)
5. Swears, E., Hoogs, A., Perera, A.G.A.: Learning motion patterns in surveillance video using hmm clustering. In: Proceedings of the IEEE Workshop on Motion and Video Computing (2008)
6. Stauffer, C.: Estimating tracking sources and sinks. In: IEEE Conference on Computer Vision and Pattern Recognition Workshop, vol. 4 (2003)
7. Stark, L., Bowyer, K.: Achieving generalized object recognition through reasoning about association of function to structure. IEEE Transactions on Pattern Analysis and Machine Intelligence 13, 1097–1104 (1991)
8. Peursum, P., West, G., Venkatesh, S.: Combining image regions and human activity for indirect object recognition in indoor wide-angle video. In: Proceedings of IEEE International Conference on Computer Vision (2005)
9. Gupta, A., Davis, L.: Objects in action: An approach for combining action understanding and object perception. In: IEEE Conference on Computer Vision and Pattern Recognition (2007)
10. Basharat, A., Gritai, A., Shah, M.: Learning object motion patterns for anomaly detection and improved object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
11. Yang, Y., Liu, J., Shah, M.: Video scene understanding using multi-scale analysis. In: Proceedings of IEEE International Conference on Computer Vision (2009)
12. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. on Patt. Analysis and Machine Intelligence 24 (2002)
13. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2 (1999)
14. Perera, A.G.A., Srinivas, C., Hoogs, A., Brooksby, G., Hu, W.: Multi-object tracking through simultaneous long occlusions and split-merge conditions. In: IEEE Conference on Computer Vision and Pattern Recognition (2006)
15. Swears, E., Hoogs., A.: Functional scene element recognition for video scene analysis. In: IEEE Workshop on Motion and Video Computing (2009)
16. Oh, S., Hoogs, A., Turek, M., Collins, R.: Content-based Retrieval of Functional Objects in Video using Scene Context. In: 11th European Conference on Computer Vision (2010)

Automatic Learning of Background Semantics in Generic Surveilled Scenes

Carles Fernández, Jordi Gonzàlez, and Xavier Roca

Dept. Ciències de la Computació & Computer Vision Center,
Edifici O, Campus UAB, 08193 Bellaterra, Barcelona, Spain
{carles.fernandez,poal,xavier.roca}@cvc.uab.es

Abstract. Advanced surveillance systems for behavior recognition in outdoor traffic scenes depend strongly on the particular configuration of the scenario. Scene-independent trajectory analysis techniques statistically infer semantics in locations where motion occurs, and such inferences are typically limited to abnormality. Thus, it is interesting to design contributions that automatically categorize more specific semantic regions. State-of-the-art approaches for unsupervised scene labeling exploit trajectory data to segment areas like sources, sinks, or waiting zones. Our method, in addition, incorporates scene-independent knowledge to assign more meaningful labels like crosswalks, sidewalks, or parking spaces. First, a spatiotemporal scene model is obtained from trajectory analysis. Subsequently, a so-called GI-MRF inference process reinforces spatial coherence, and incorporates taxonomy-guided smoothness constraints. Our method achieves automatic and effective labeling of conceptual regions in urban scenarios, and is robust to tracking errors. Experimental validation on 5 surveillance databases has been conducted to assess the generality and accuracy of the segmentations. The resulting scene models are used for model-based behavior analysis.

1 Introduction

The automatic analysis of human behaviors in large amounts of video surveillance footage is becoming critical as the number of cameras installed in public areas increases. This demand has generated novel techniques for the analysis of large collections of archives containing recordings from different outdoor scenarios during long periods. As a result, events of interest are detected, and alarms can be raised online according to predefined criteria [1]. Complementary, events extracted from image sequences can be used for annotation purposes when becoming concepts to index surveillance databases.

There is a clear trade-off between the semantic richness of video events and the robustness of their recognition. The richness of the conceptual knowledge extracted from surveillance sequences greatly determines the limitations of eventual user queries. Ideally, indexing would be based on high-level concepts determined by rich and complete ontologies [2]. However, as events become more specific, their recognition in surveillance data also becomes more challenging.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 678–692, 2010.

Important steps forward have been taken in the computer vision domain, where interesting approaches appeared related to the automatic interpretation of human activities within scenes. In surveillance data obtained from static cameras in outdoor scenes, human activities are commonly represented by trajectories of points extracted using detection and/or tracking processes.

On the one hand, different statistical properties of these observed trajectories are computed in order to assess their normal or abnormal nature. There are several strategies to cluster and merge trajectories, like spatial extension [3], Hierarchical Fuzzy C-Means [4], Hierarchical clusters [5], GMMs [6], or splines [7], among others. Subsequently, by analyzing the regions where motion is observed, characteristic regions like roads, walking paths, or entry/exit points can be learned [8]. Statistical techniques have been also used to model semantic regions based on activity correlation [9]. These robust bottom-up processes are scene-independent, and abnormal behaviors like violent struggling can be detected and annotated, e.g., by observing erratic trajectories with high speed variations.

On the other hand, deterministic models provided beforehand by an expert have been also applied successfully in the surveillance domain, such as Situation Graph Trees [10,11], Petri Nets [12], or Symbolic Networks [13], for example. These models can represent complex behaviors (such as *'danger_of_runover'* or *'car_overtaking'*) while performing reasoning based on more simple, but robustly detected, events (e.g., *'turning_left'* or *'accelerating'*), for example those ones extracted using the aforementioned bottom-up processes. Hence, high-level reasoning processes can generate key-words and concepts associated to stronger semantics that can be searched for.

Towards this end, reasoning on events requires of conceptual scene models that semantically represent the background of the surveilled scene. The semantics of the regions in which an agent is found at each time step are used to infer behaviors, such as *'crossing_the_street'* or *'waiting_at_the_crosswalk'*. Unfortunately, each particular scene requires of its own conceptual scene model. Therefore, there is a need for automatic and generic learning procedures able to infer the semantics of thousands of surveillance scenes.

In this paper we present a novel technique for automatic learning of conceptual scene models using domain knowledge, which can be successfully used for further reasoning and annotation of generic surveillance sequences. In essence, we learn spatiotemporal patterns of moving objects to infer the semantic labels for background regions where motion has been observed, such as pedestrian crossings, sidewalks, roads, or parking areas. The derivation of site labels is formulated as a MAP-MRF inference in terms of a pairwise Markov network, whose graph configuration is factored into a joint probability guided by taxonomical knowledge. Finally, we have applied the SGT formalism to demonstrate the applicability of our approach, although other deterministic behavior models that require of conceptual scene models can be used instead.

This paper is structured as follows: Section 2 formally defines our labeling task in terms of maximization of region compatibility. It comprises two steps: the compatibility with observed evidence is computed from motion features in

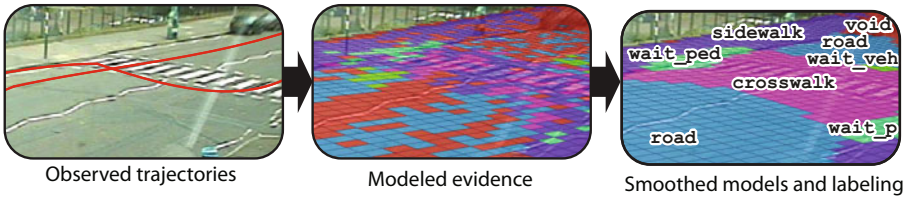


Fig. 1. Scheme of the proposed framework for labeling urban scenarios

Section 3, and inter-region compatibility for smoothness is modeled in Section 4. A preprocessing stage to improve efficiency is explained in Section 5. The method is tested thoroughly for experimental validation in Section 6, and applied to the SGT formalism in Section 7. Finally, we discuss the results and provide some concluding remarks.

2 Background Labeling by Compatibility

The semantic learning of a background model consists of partitioning an arbitrary scenario of the domain into a lattice of regions, and have each region learn a spatiotemporal model. Each model should be estimated based on trajectory properties, and finally assigned an explicit label that categorizes it. Here, we tackle the problem of *semantic region learning* as one of *multiclass semantic segmentation*. Towards this end, efficient techniques have been developed, such as MRF [14] and its variants, like DRF [15], or LCRF [16], or alternatives like Semantic Textons [17]. In our case, the categorization of regions from their statistical models will be posed as a labeling task and formulated as a MAP-MRF inference problem, defined by irregular sites and discrete labels [18].

2.1 Sites and Labels

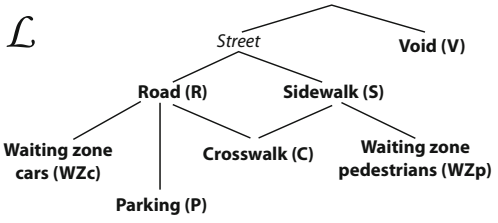
The lattice of irregular regions to be labeled is usually defined either by perceptual groups –out of a segmentation process–, or by clusters of recognized features within the scene [18]. Instead, we aim to define lattices that capture the condition of far-field projectivity, characteristic of scenarios in our domain.

To do so, we compute the scene to ground-plane homography [19], so that each lattice is a set of regions $\mathcal{R}$ obtained as the projection of a rectangular grid from ground-plane to scene. In addition to the sites, a set $\mathcal{L}$ of seven discrete labels defines generic, common, and relevant locations in urban surveillance. Labels are organized taxonomically as shown in Table 1a. A void label (V) is made available for those cases in which none of the labels applies, as in [20].

2.2 Inference

Having defined the set of sites and labels, we next describe the process of assigning a label $l \in \mathcal{L}$ to each region $r \in \mathcal{R}$. The disparity of labels is assumed

Table 1. (a) Taxonomy of locations for urban surveillance. (b) Each location is a vector of the trinary features *ped*=Pedestrian, *veh*=Vehicle, *wai*=Wait, and *stp*=Stop.



(a)

Label from $\mathcal{L}$		ped	veh	wai	stp
C	$f^1 = [$	+1	+1	0	-1]
S	$f^2 = [$	+1	-1	-1	-1]
R	$f^3 = [$	-1	+1	-1	-1]
WZp	$f^4 = [$	+1	-1	+1	-1]
WZc	$f^5 = [$	-1	+1	+1	-1]
P	$f^6 = [$	0	+1	0	+1]
V	$f^7 = [$	-1	-1	-1	-1]

(b)

to be piecewise smooth in the lattice of regions. A series of observation vectors $o = \{x, y, a\}$ constitutes the evidence from the trajectories, where (x, y) is the estimated position of the agents in the image plane –the centroid of the ellipsoid projected to the ground-plane–, and a is a binary parameter stating whether the agent is a vehicle or a pedestrian. The derivation of the site labels $\{l\}$ is formulated as a MAP-MRF inference in terms of a pairwise Markov network, whose graph configuration is factored into the joint probability

$$P(\{l\}, \{o\}) = \frac{1}{Z} \prod_{r \in \mathcal{R}} \phi_r(l_r, o_r) \prod_{\{r,s\} \in \mathcal{N}} \psi_{r,s}(l_r, l_s), \tag{1}$$

where Z is a normalization factor. The *data compatibility* function $\phi_r(l_r, o_r)$ is interpreted as the likelihood of choosing label l for region r, given o observed in r. This function is learned by trajectory analysis as explained in Section 3.

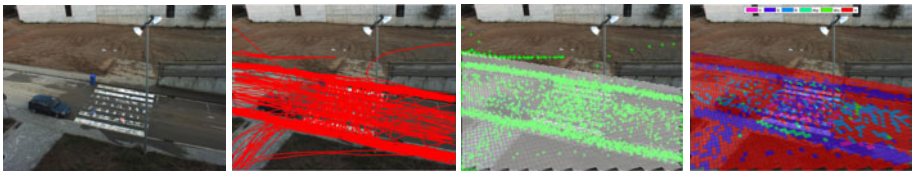
On the other hand, smoothness constraints are encoded into $\psi_{r,s}(l_r, l_s)$, so-called *internal binding*, which models how neighboring regions affect to each other regarding their classes. In this term, the set $\mathcal{N}$ contains all pairs of interacting regions, in our case adjacent 8–connected regions in the projected grids. In our work, $\psi_{r,s}(\cdot)$ is a prior set of constraints directly taken from topological assumptions that are derived from a defined hierarchy of labels depicting domain knowledge, as later explained in Section 4.

Once $\phi_r(\cdot)$ and $\psi_{r,s}(\cdot)$ are defined, a max-product belief propagation (BP) algorithm [20] derives an approximate MAP labeling for Eq. (1).

3 Data Compatibility

We define the function $\phi_r(l_r, o_r)$ as the likelihood of region r to be labeled as l, having observed a series of vectors o_r in the region, and according to a motion-based model that encodes prior domain knowledge.

Challenges arisen by semantic scene –similarly, by document analysis or medical imaging– deal with overlapping classes that are not mutually exclusive. Hence, we characterize scenario regions following the prototype theory, which defines class labels as conjunctions of required (*+1*), forbidden (*-1*), and irrelevant (*0*) features [21]. Here, labels are modeled using 4 features: target (i) is a



(a) (b) (c) (d)

Fig. 2. Region modeling by trajectory analysis: (a) original image, (b) smoothed trajectories, (c) sampled control points, (d) initial labeling

pedestrian or (ii) a *vehicle*, (iii) is *waiting*, and (iv) has *stopped*, see Table 1b. A series of prototypical feature vectors $\{f^1 \dots f^{|\mathcal{L}|}\}$ results from this step.

Next step consists of online smoothing and sampling data from tracking. To do so, each new complete trajectory is fitted by iteratively increasing a sequence of connected cubic b-splines [7], see Fig. 2b: an adjustment step divides a spline into connected sub-splines more fitted to the trajectory, and a termination step validates a subsequence when its maximum distance to the trajectory is below a 10% of the total length. Once the recursion is done, the global sequence of splines is sampled to generate time-equidistant control points (Fig 2c), each one having an observation $o = \{x, y, a\}$. The position (x, y) is estimated by a multi-target tracker [22], and the target type (a) is identified using a scene-invariant discriminative approach as in [23]. When a new control point is generated, its enclosing region updates an histogram of the 4 features described above. Lastly, each region is assigned an online averaged vector of observed features f_o.

The data compatibility of the observations in region r with label $l \in \mathcal{L}$ is a softmax function of the Hamming distance between the averaged vector of features observed, f_o, and the vector defined for that label, f^l:

$$\phi_r(l_r, o_r) = \frac{\exp(-d_H(f_o, f^l))}{\sum_{m \in \mathcal{L}} \exp(-d_H(f_o, f^m))}. \tag{2}$$

Learned data compatibilities provide an initial rough scene model. This initial labeling omits the inference phase, and simply assigns to each region the label with a highest value of $\phi_r(\cdot)$, see Fig. 2d. Due to the limited coverage of the scene by the control points, there is a massive presence of *Void* labels, in red.

4 Smoothness

The smoothness term $\psi_{r,s}(l_r, l_s)$ specifies inter-region compatibilities, stating how the system privileges or disfavors label l_r at expenses of l_s when r and s are adjacent. In other words, it conditions *a priori* those neighborhoods formed by a certain combination of semantic categories. The goal here is to specify compatibilities that discard unlikely labelings, smooth poorly sampled ones, and preserve detailed information that are scarce but consistent.

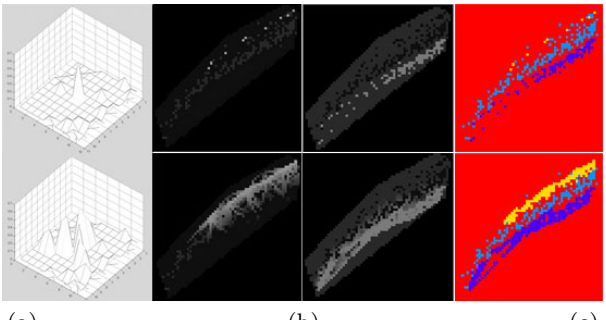
In our case, advantage is taken on the hierarchical organization of $\mathcal{L}$ to constrain discontinuities between labels. $\mathcal{L}$ fixes topological constraints of set inclusion, by establishing relations of particularization as seen in Table 1a; e.g., a *parking* is a concrete segment of *road*, and also constrains the adjacency between different regions. Consequently, compatibilities are fully specified by

$$\psi_{r,s}(l_r, l_s) = \begin{cases} 1 & l_r = l_s \\ \alpha & Adj(l_r, l_s) \\ \beta & \text{otherwise} \end{cases} \tag{3}$$

where $1 > \alpha > \beta > 0$, and $Adj(l_r, l_s)$ states that l_r and l_s are adjacent in the topological map, i.e., have direct links in the taxonomy. For example, P–R, C–R or C–S are adjacent pairs, but WZc–P or R–S are not. This model firstly maintains the identity of the labels, secondly favors dilation and erosion between adjacent regions, and ultimately allows relabeling for region smoothness.

5 Geodesic Interpolation

Having defined compatibilities for observed evidence and sought smoothness, the application of an efficient BP algorithm [20] approximates an optimal labeling via MAP-MRF inference. Nonetheless, In cases of very poor sampling, e.g., when estimating models of parkings, the regions obtained by MAP-MRF inference with the smoothness prior are often still disconnected or not representative, making it difficult to obtain accurate segmentations.



(a) (b) (c)

Fig. 3. Top: non-smoothed marginal probabilities viewed (a) as a discrete mesh and (b) as intensity maps, and (c) initial label assignment (best viewed in color). Bottom: effects of the interpolation.

To solve this problem, a preprocessing stage is used before the inference step to reinforce spatial coherence by interpolating lines in a geodesical manner. The idea is to create linear ridges that connect high-valued and isolated samples in each label's marginal probability map (Fig. 3a), in order to emphasize the

presence of connected structures in them (Fig. 3b). As a result, the subsequent MAP-MRF process is reinforced with these structures and guides more sensible inferences for an eventual labeling, as shown in Fig. 3c.

The algorithm recursively finds non-void assigned categories that are isolated, i.e., have no neighbor with the same category. Regions with the same class assignment are searched within an area of influence –we used 1.5 meters in the calibrated map–, and the two regions are connected with a linear ridge, which modifies the marginal class of the regions on that line only if the original marginal value increases. The class probabilities of each region are finally normalized for each region, and new assignments are applied.

6 Evaluation

The presented framework has been evaluated in 5 urban scenarios with diverse characteristics, obtained from camera recordings. The *Hermes* dataset[1] presents an interurban crosswalk scenario with more pedestrians than vehicles; *Oxford centre*[2] shows an intersection highly populated by both target types; *Devil's Lake*[3] presents moderate agitation but challenges with an intense projectivity; *Kingston–1* contains a partially seen bus stop close to a crosswalk, and *Kingston–2* shows a minor street with perpendicular parking spaces used for long periods of time. These two last scenarios are extracted from the Kingston dataset [24].

Evaluation is carried out using 25 ground truth (GT) images –5 participants per scenario–, consisting of pixel-level maps segmented into the 7 categories of Table 1. Participants were asked to visually identify the semantic regions by observing recorded footage, and partition them accordingly. In order to evaluate discriminant capability, and given that manual labeling is prone to vary across humans, the system will perform well if segmentation errors compare to inter-observer variability. This validation criterion is commonly used in biometrics [25]. To accomplish this, each GT image has been divided into the cells of its corresponding grid, and a modal filter has been applied over each cell, assigning the most repeated pixel label to that region. Finally, each label assignment has been evaluated against the other GTs and averaged for each GT and scenario.

In order to obtain quantitative comparisons, we have computed 3 different accuracy scores over the 5 datasets, evaluating both techniques against the GT assignments. In the evaluation tests, the maximum number of iterations for the MAP-MRF has been limited to 15. The values of α and β are 0.8 and 0.6 respectively, for all experiments.

The matricial configuration of the lattice reduces computational effort in both region modeling and label inference. Observations update the region models online as trajectories are complete. Regarding the final inference over regions learned, for a grid size of 75×75 geodesic interpolation takes at most 3 seconds to

[1] http://www.hermes-project.eu/
[2] http://webcam.oii.ox.ac.uk/
[3] http://www.gondtc.com/web-cams/main-street-large.htm

Table 2. Number of correctly tracked (a) pedestrians and (b) vehicles in each scenario, and amount of observation errors due to: (c) agent misclassification, (d) lost or missed tracks, and (e) false detections

	Correct			Erroneous			
Scenario (total tracks)	(a)	(b)	Total	(c)	(d)	(e)	Total
Hermes (161)	103	26	129	13	10	9	32
Oxford centre (180)	87	62	149	20	8	3	31
Devil's Lake (179)	49	98	147	17	10	5	32
Kingston–1 (161)	85	53	138	12	9	2	23
Kingston–2 (87)	35	33	68	7	4	8	19

complete, and the BP algorithm with maximum iterations takes approximately 90 seconds in a Pentium II 3 GHz machine with 2 Gb RAM.

We analyze the consistency of the results by testing over a wide range of grid size values, which is the main parameter involved in the sampling process: given that each control point sampled from a trajectory affects uniquely its enclosing region, the number of cells tesselating the scenario is indicative of the area of influence of tracked objects during region modeling. The dimensions of the projected grid in our experiments range from 40×40 to 150×150. Lower cell resolutions do not capture the details of the scenario, thus not being suitable to model semantic regions. Greater resolutions show performances that are similar to the displayed range, but require substantially more computational resources.

Additionally, the tracked trajectories used as observations incorporate errors. Each error consists of one or more of the following cases: misclassification of agents, lost tracks, and false detections. Table 2 gives numerical information on the agents involved in each scenario and the number and type of erroneous observations. The system has been evaluated with and without the presence of errors, in order to test its robustness.

6.1 Quality Scores

Performances have been evaluated in terms of accuracy. Three scores have been considered: overall accuracy (OA), segmentation accuracy (SA), and weighted segmentation accuracy (WSA). The two former scores are defined by

$$OA = \frac{TP+TN}{TP+FP+TN+FN}, \qquad SA = \frac{TP}{TP+FP+FN},$$

where OA is traditional accuracy, typically overfavored in multiclass contexts given the high value of True Negatives as the number of classes increments. For this reason, SA has been increasingly used to evaluate multiclass segmentations, as in the PASCAL-VOC challenge [4]. Additionally, WSA is defined by

$$WSA = \frac{TP^*}{TP^*+FP^*+FN^*},$$

[4] http://pascallin.ecs.soton.ac.uk/challenges/VOC/

in which an assignment is now considered true positive if the inferred label is either equal to the ground truth, or is its direct generalization in the taxonomy of Table 1a; and negative otherwise, thus modifying the account of errors. For instance, an actual *parking* is here positively labeled as *road*, and a *pedestrian waiting zone* is correctly labeled as *sidewalk*. Note that this score does not necessarily benefit our approach, since our smoothness constraints do not award class generalization. Instead, the goal of this metric is to penalize wrong particularizations. GT evaluation in Fig. 4a shows that WSA takes into account consistency in different GT realizations –unlike SA–, while penalizing differences harder than OA.

6.2 Median Filter

We have compared our method to median filters. They are the most used nonlinear filters to remove impulsive or isolated noise from an image, a typical type of noise found in our problem domain. Median filters preserve sharp edges, which makes them more robust than traditional linear filters and a simple and cheap solution to achieve effective non-linear smoothing. They are commonly used for applications of denoising, image restoration, and interpolation of missing samples, all of which are applicable in our context.

We have compared the performances obtained by a median filter after 15 iterations and by our proposed inference framework, to evaluate the contributions of taxonomy-based constraints to the smoothing task. The filter is applied for each marginal probability map $P(f_r = l), l = 1 \dots |\mathcal{L}|$, maintaining the MRF neighborhood defined. A median-filtered labeling is performed by assigning the most probable label to each region, once the process has converged or exceeded the maximum number of iterations allowed.

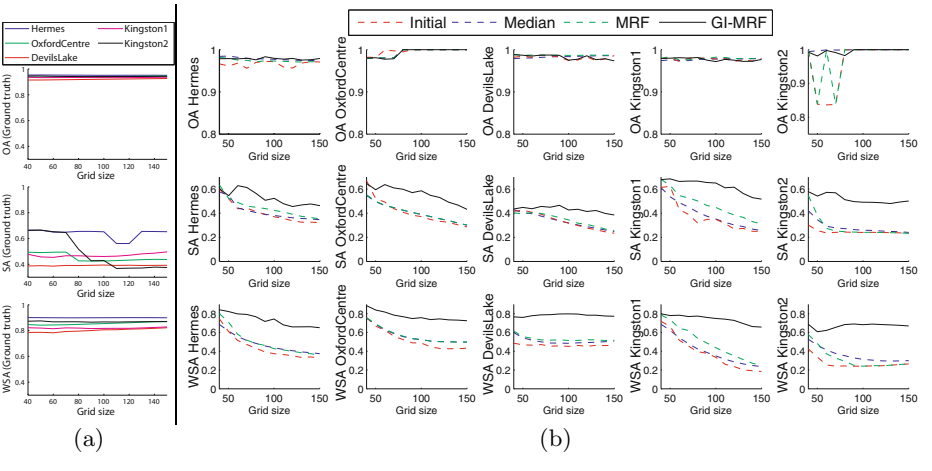


Fig. 4. (a) Evaluation of the inter-observer variability in GT segmentations. (b) Statistical scores for the 5 considered scenarios. More details in the text.

Table 3. Quantitative OA, SA, and WSA scores for a grid size of 75×75, without and with the presence of erroneous trajectories

		Overall accuracy (OA)				Segmentation accuracy (SA)				Weighted segmentation accuracy (WSA)			
		Initial	Median	MRF	GI–MRF	Initial	Median	MRF	GI–MRF	Initial	Median	MRF	GI–MRF
Only correct	Hermes	0.98	0.96	0.97	0.98	0.40	0.40	0.45	**0.64**	0.50	0.44	0.51	**0.77**
	Oxford Centre	0.98	0.97	0.98	0.98	0.46	0.52	0.58	**0.61**	0.65	0.66	0.75	**0.93**
	Devil's Lake	0.98	0.99	0.99	0.99	0.37	0.39	0.39	**0.44**	0.49	0.46	0.52	**0.78**
	Kingston–1	0.98	0.97	0.98	0.99	0.43	0.37	0.50	**0.66**	0.46	0.44	0.59	**0.76**
	Kingston–2	1.00	0.84	1.00	0.98	0.27	0.24	0.28	**0.56**	0.36	0.24	0.35	**0.69**
	Average	0.98	0.94	0.98	0.98	0.39	0.38	0.44	**0.58**	0.49	0.45	0.54	**0.78**
Correct and erroneous	Hermes	0.98	0.97	0.97	0.98	0.40	0.40	0.45	**0.53**	0.51	0.45	0.52	**0.78**
	Oxford Centre	0.98	0.97	0.98	0.98	0.46	0.53	0.56	**0.57**	0.66	0.68	0.76	**0.94**
	Devil's Lake	0.98	0.99	0.99	0.99	0.37	0.39	0.40	**0.43**	0.50	0.47	0.53	**0.78**
	Kingston–1	0.97	0.98	0.98	0.98	0.43	0.40	0.50	**0.65**	0.46	0.50	0.60	**0.76**
	Kingston–2	1.00	0.84	0.99	0.98	0.28	0.24	0.34	**0.55**	0.38	0.26	0.40	**0.76**
	Average	0.98	0.95	0.98	0.98	0.39	0.39	0.46	**0.55**	0.50	0.47	0.56	**0.80**

6.3 Results

Fig. 4a shows the results of the inter-observer evaluation for the GT, which constitute the baseline of the system's performance. Fig. 4b shows quantitative scores for OA, SA, and WSA in the 5 scenarios. Each plot draws the results of 4 different approaches, applied to the 5 series of GT available. These approaches correspond to: (i) assigning labels using only observed evidence from trajectories, i.e., neglecting smoothness priors (*Initial*); (ii) using a median filter over the initial models (*Median*); (iii) applying MAP-MRF inference (Eq. 1) to the initial models (*MRF*); and (iv) applying a preprocessing step based on geodesic interpolation to the region models (*GI–MRF*).

Results are similar to GT inter-observer variability. Only occasional plot oscillations appear in Kingston2 for the OA measure, due to the non-linear operation of sampling GT images into lattices of a concrete size. Moreover, increasing the cell resolution progressively lowers the quality of the initial models, as well as the accuracy on posterior labelings. Nonetheless, it is shown that interpolation grants a performance almost invariant to the grid size used. This is emphasized in case of poor sampling, e.g, parking spaces.

Table 3 shows numerical results for a grid of 75 × 75 cells, with and without noisy trajectories. As seen in this table, OA is excessively favored due to the high number of true negatives produced in a multiclass context, thus suggesting SA and WSA as more convenient to compare the different techniques. Particularly, WSA should be interpreted as the precaution to avoid wrong particularizations. With these metrics, experiments using geodesic interpolation and smoothness constraints practically always achieve the maximum score, whereas a median filter fails dramatically as the grid resolution increments, or in case of

ill-convergence, e.g., it fails to preserve parking regions in *Kingston–2*. Additionally, it is seen that even by incorporating erroneous trajectories to the datasets, letting them be about a 20% of the total, the accuracy values remain stable.

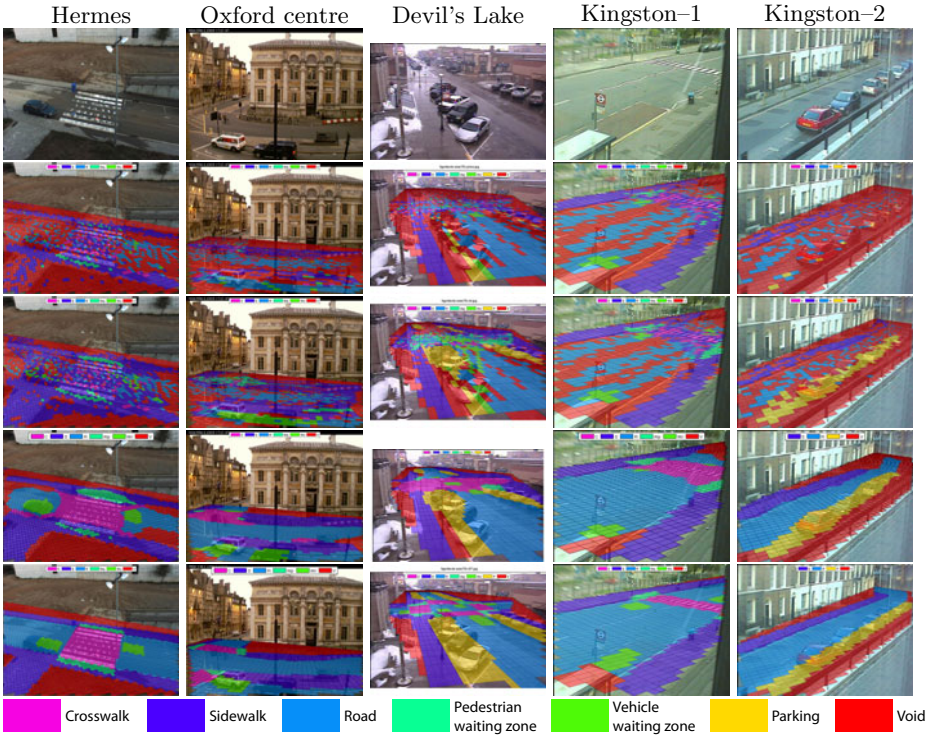


Fig. 5. Labeling step results for a 75×75 grid: First row shows original image, second row the initial labeling only from observations, third row the labeling with geodesic interpolation, fourth row the inference labeling using both interpolation and smoothness constraints, and the bottom row shows the GT. Best viewed in color.

Fig 5 depicts qualitative step results of the labeling process for a grid size of 75×75. For visualization purposes, results are shown within a ROI. The depicted results represent the activity of the tracked objects, rather than the visual appearance of the scenario. Instead, appearance is commonly used to guide manual labelings. We also identify an edge-effect of *Void* regions, given that control points near the edges often lack of precedent or consecutive samples to update their regions. This happens especially for vehicles, due to their higher speed and poorer sampling. Finally, cases of intense projectivity –e.g., *Devil's Lake*–, make it more difficult for the models to emphasize the presence of connected regions, thus provoking generalized smoothing.

7 Application

Finally, the conceptual scene model have been used to exploit model-based behavior analysis. This has been achieved using the Situation Graph Tree (SGT) [10] shown in Fig. 6(a,b), although any symbolic approach requiring conceptual scene models could be used instead, like Petri Nets or Symbolic Networks. We choose

Fig. 6. SGT used to interpret behaviors of (a) vehicles and (b) pedestrians. (c,d) Selected frames from each interval. Semantic predicates are generated deterministically using (e) the learned region maps and (f) their corresponding GT maps.

SGTs because they reason about the events observed in the learned semantic regions, and can annotate situations of interest and traffic behaviors.

The scenario-independent SGT used generates conceptual descriptions when certain conditions happen, such as vehicles entering sidewalks or pedestrians entering roads. In addition, basic interpretations are formulated; e.g., if a vehicle stops in front of a crosswalk where a pedestrian is found, it is *giving_way* to this person; and if a vehicle stops in a parking, it has *parked* there. In essence, basic conceptual predicates are inferred by a *fuzzy metric-temporal reasoner*, we refer the reader to [10,11] for implementation details.

Fig. 6 shows predicates generated in *OxfordCentre* and *DevilsLake* at different time intervals. Most frequently, the generated predicates differ only at the beginnings or endings of the temporal intervals; this is due to slight variations among region boundaries. In Fig 6c, two predicates from the left column are not found in the right one, since a *WZc* zone has not been identified in the GT model. Nevertheless, alarms and simple interpretations are correctly generated.

8 Conclusions

We have shown an effective motion-based method to automatically label semantic zones. The method has been applied to different urban scenarios using the same behavioral models. Our approach enhances state-of-the-art on background labeling by using taxonomical knowledge to guide consistent inferences during labeling. It is scene-independent, viewpoint-invariant and of reduced computational cost, for it does not require to compute costly image descriptors.

Initial region models are learned from trajectory features, and updated as new trajectories are available. Smoothness is taken into account using a MAP-MRF inference, whose parameters are conditioned by prior taxonomical domain knowledge. The framework is scenario-independent: it has been applied to 5 datasets showing different conditions of projectivity, region content and configuration, and agent activity. Step results are shown at every stage of the process, to capture the particular contributions of each task. The method has been compared to a median filter, showing its better performance on the 3 scores tested.

Our work makes it possible to use predefined behavior models in generic surveillance scenes. By automatically learning the conceptual scene model behind lots of outdoor scenes, we can evaluate existing deterministic models (SGT, Petri Nets, Symbolic Networks) in terms of generalization or scaling criteria. Further steps include improving the accuracy of inter-region boundaries and extending the system to indoor scenarios. Such environments incorporate more complex semantics on agent actions and interactions, so deterministic behavior models using domain knowledge can be used to extract key concepts for annotation.

Acknowledgements

This work has been supported by the Spanish Research Programs Consolider-Ingenio 2010:MIPRCV (CSD200700018) and Avanza I+D ViCoMo

(TSI-020400-2009-133); and by the Spanish projects TIN2009-14501-C02-01 and TIN2009-14501-C02-02. We acknowledge the valuable collaboration of Dr. Pau Baiget.

References

1. Robertson, N., Reid, I.: A general method for human activity recognition in video. CVIU 104, 232–248 (2006)
2. Ballan, L., Bertini, M., Serra, G., Del Bimbo, A.: Video annotation and retrieval using ontologies and rule learning. IEEE Multimedia (2010)
3. Makris, D., Ellis, T.: Learning semantic scene models from observing activity in visual surveillance. IEEE TSCM, Part B 35, 397–408 (2005)
4. Hu, W., Xiao, X., Fu, Z., Xie, D.: A system for learning statistical motion patterns. PAMI 28, 1450–1464 (2006)
5. Piciarelli, C., Foresti, G.L.: On-line trajectory clustering for anomalous events detection. PRL 27, 1835–1842 (2006)
6. Basharat, A., Gritai, A., Shah, M.: Learning object motion patterns for anomaly detection and improved object detection. In: CVPR, Anchorage, USA (2008)
7. Baiget, P., Sommerlade, E., Reid, I., Gonzàlez, J.: Finding prototypes to estimate trajectory development in outdoor scenarios. In: 1st THEMIS, Leeds, UK (2008)
8. 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)
9. Li, J., Gong, S., Xiang, T.: Scene segmentation for behaviour correlation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 383–395. Springer, Heidelberg (2008)
10. Nagel, H.H., Gerber, R.: Representation of occurrences for road vehicle traffic. AI-Magazine 172, 351–391 (2008)
11. Gonzàlez, J., Rowe, D., Varona, J., Xavier Roca, F.: Understanding dynamic scenes based on human sequence evaluation. IVC 27, 1433–1444 (2009)
12. Albanese, M., Chellappa, R., Moscato, V., Picariello, A., Subrahmanian, V.S., Turaga, P., Udrea, O.: A constrained probabilistic petri net framework for human activity detection in video. IEEE TOM 10, 982–996 (2008)
13. Fusier, F., Valentin, V., Bremond, F., Thonnat, M., Borg, M., Thirde, D., Ferryman, J.: Video understanding for complex activity recognition. MVA 18, 167–188 (2007)
14. Kumar, M., Torr, P., Zisserman, A.: Obj. Cut. In: CVPR (2005)
15. Kumar, S., Hebert, M.: Discriminative fields for modeling spatial dependencies in natural images. In: Advances in Neural Information Processing Systems, vol. 16 (2004)
16. Winn, J., Shotton, J.: The layout consistent random field for recognizing and segmenting partially occluded objects. In: CVPR, pp. 37–44 (2006)
17. Shotton, J., Johnson, M., Cipolla, R., Center, T., Kawasaki, J.: Semantic texton forests for image categorization and segmentation. In: CVPR (2008)
18. Li, S.: Markov random field modeling in image analysis. Springer, Heidelberg (2001)
19. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge Univ. Press, Cambridge (2003)
20. Felzenszwalb, P., Huttenlocher, D.: Efficient belief propagation for early vision. IJCV 70, 41–54 (2006)

21. Croft, W., Cruse, D.: Cognitive linguistics. Cambridge Univ. Press, Cambridge (2004)
22. Rowe, D., Gonzàlez, J., Pedersoli, M., Villanueva, J.: On tracking inside groups. Machine Vision and Applications 21, 113–127 (2010)
23. Bose, B., Grimson, E.: Improving object classification in far-field video. In: CVPR (2004)
24. Black, J., Makris, D., Ellis, T.: Hierarchical database for a multi-camera surveillance system. Pattern Analysis and Applications 7, 430–446 (2004)
25. Landis, J., Koch, G.: The measurement of observer agreement for categorical data. Biometrics 33, 159–174 (1977)

Why Did the Person Cross the Road (There)? Scene Understanding Using Probabilistic Logic Models and Common Sense Reasoning

Aniruddha Kembhavi, Tom Yeh, and Larry S. Davis

University of Maryland, College Park
anikem@umd.edu, tomyeh@umiacs.umd.edu, lsd@cs.umd.edu

Abstract. We develop a video understanding system for scene elements, such as bus stops, crosswalks, and intersections, that are characterized more by qualitative activities and geometry than by intrinsic appearance. The domain models for scene elements are not learned from a corpus of video, but instead, naturally elicited by humans, and represented as probabilistic logic rules within a Markov Logic Network framework. Human elicited models, however, represent object interactions as they occur in the 3D world rather than describing their appearance projection in some specific 2D image plane. We bridge this gap by recovering qualitative scene geometry to analyze object interactions in the 3D world and then reasoning about scene geometry, occlusions and common sense domain knowledge using a set of meta-rules. The effectiveness of this approach is demonstrated on a set of videos of public spaces.

Keywords: Scene Understanding, Markov Logic Networks.

1 Introduction

We build on recent research in appearance-based object recognition and tracking [1,2,3,4], recovery of qualitative scene geometry from images and video [5,6,7], and probabilistic relational models for integrating common sense domain models with uncertain image analysis [8], to develop a video understanding system that can identify scene elements (cross walks, bus stops, traffic intersections), characterized more by qualitative geometry and activity than by intrinsic appearance. The domain models we use are naturally specified by humans, and characterize scene elements in terms of geometric relationships (sidewalks are found along roads and are parallel to roads) and activity relationships (people walk on sidewalks, wait and possibly queue for a bus).

These domain models are related to image analysis (appearance, tracking, motion) by representing them as probabilistic logical models (Markov Logic Networks). These logical models, however, describe *what typically happens* in the scene and not *what is visible* in some video of that scene. We bridge this gap using two methods. First, we recover qualitative scene geometry to analyze object interactions in the 3D world rather than the 2D image plane. Second,

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 693–706, 2010.

we utilize a set of meta-rules that capture general rules about scene geometry and occlusion reasoning and fuse them with common sense domain knowledge to detect these scene elements in videos taken from arbitrary viewpoints. This involves reasoning about unobserved events and inferring their occurrence based on other observations. Figure 1 provides an overview of our system.

Fig. 1. System overview. Our scene understanding system consists of an image analysis module (Section 3) that takes an input video and outputs a set of events and zone characteristics as observational evidence, a knowledge base (Section 4) that stores human elicited domain models and general rules about scene geometry and occlusion as a set of first-order logic rules, and an inference engine (Section 5) based on Markov Logic Networks that uses the logic rules and observational evidence to infer the labels of visible scene elements.

As an example, consider a model for a bus-stop. This model might indicate that people wait and queue at a bus stop, a bus stops at the bus stop, the doors to the bus open, people leave the bus through the doors, then the people waiting enter the bus through the doors, the doors close, and finally the bus leaves. From the viewpoint in Scenario 1 (refer to Figure 2), all of the activities associated with this bus stop model are observable. Scenario 2 shows a bus stop seen from another viewpoint, in which the bus occludes the people waiting to board, and the bus doors are not visible. In this case, our system reasons about this occlusion, and determines that what we expect to observe are that the people waiting for the bus will be gone when the bus leaves, and that new people will be seen after the bus leaves.

Fig. 2. Two bus stops observed from different viewpoints. In Scenario 1, all activities associated with a typical bus stop model are observable. In Scenario 2, the bus occludes people departing and entering the bus.

We demonstrate our video understanding framework on a dataset of videos of public spaces. These video sequences were collected using cameras overlooking scenes from varying viewpoints. Each contains multiple scene elements of interest, such as bus stops, traffic intersections, stop signs, crosswalks, garage entrances, etc. Our system is able to correctly identify a large number of these scene elements described by the human elicited domain models.

2 Related Work

Methods to categorize scenes from single images by completely bypassing the tasks of image segmentation and object detection are described in [9,10,11]. Oliva et al. [9] represented holistic image structure using low level features that captured the degree of naturalness, openness, ruggedness, etc. whereas Fei-Fei et al. [10] represented scenes as bags of codewords of texture measures. More

recently, there have been attempts to jointly solve the tasks of object recognition and scene classification [12,13,14,15]. Bosch et al. [13] detected objects and then used the object distribution for scene classification. Murphy et al. [14] combined the holistic image representation of [9] with local object detectors using a tree-structured graphical model. Li et al. [15] proposed a framework to deal with three problems simultaneously: object detection, segmentation and scene categorization.

There has also been progress in recovering surface orientations [5,7] and occlusion boundaries [16], given just a single image. Recently, Hoiem et al. [17] proposed a framework in which estimates of surface orientations, occlusion boundaries, objects, camera viewpoint and relative depth are combined, enabling automatically reconstructed 3D models.

Research in the domain of scene understanding from videos has mostly focused on building models of motion patterns of objects and using these to detect anomalous behaviors [18,19,20,21]. While Hu et al. [20] propose a parametric approach to model typical scene behaviors, Saleemi et al. use non-parametric density functions. Building such typical behavior models can help to improve foreground detection, detect areas of occlusion and identify anomalous motion patterns. There have also been attempts to learn activity based semantic region models for locations such as roads, paths, and entry/exits, most notably by Makris et al. [19] and Swears et al. [22]. Both these approaches involved designing a detector for every scene element.

Research in object category recognition has typically focused on building visual classifiers trained on annotated datasets. Recently however, there has been a growing interest in building object category models directly from human elicited descriptions [23,24,25]. Such approaches have the potential to learn unseen object categories based on their descriptions in terms of known visual attributes.

3 Image Analysis

Our scene understanding framework has three components: an image analysis module, a knowledge base and an inference module (refer to Figure 1 for a system overview). The image analysis module first segments the scene into a set of neighborhoods called *zones*. It then analyzes appearance characteristics of each zone as well as motion properties of objects passing through them, to generate a set of zone attributes that characterize local scene geometry and capture occlusion relationships between zones. A set of dynamic events is then generated for every zone, at every time instant, to describe the behavior of objects in the scene. The knowledge base consists of domain models describing the scene elements of interest, as well as a set of meta-rules that capture general knowledge about scene geometry and occlusion. The inference module, based on Markov Logic Networks (MLN), integrates events generated by the image analysis component with the rules in the knowledge base to label scene elements. The knowledge base and inference module are described in Sections 4 and 5 respectively. The components of the image analysis module are described below.

3.1 Detection and Tracking

We detect and track three classes of objects: humans, cars and buses. Detection is carried out using the object detection method proposed in [2][1]. For the purposes of human detection, we directly used a trained model provided along with the code, which was trained on the INRIA pedestrian dataset [1]. The car detector is trained using the Caltech Car Rear Training Set and the ETHZ Car Side Training Set [26]. The bus detector is trained using images from Bing Image Search. A two level association based tracking method is used to link object detections into tracks. At the low level, detections are linked to form tracklets using appearance and proximity features. At the second level, these tracklets are associated into longer tracks using appearance and motion features. Figure 3b shows car and human tracks obtained for one of the videos in our dataset.

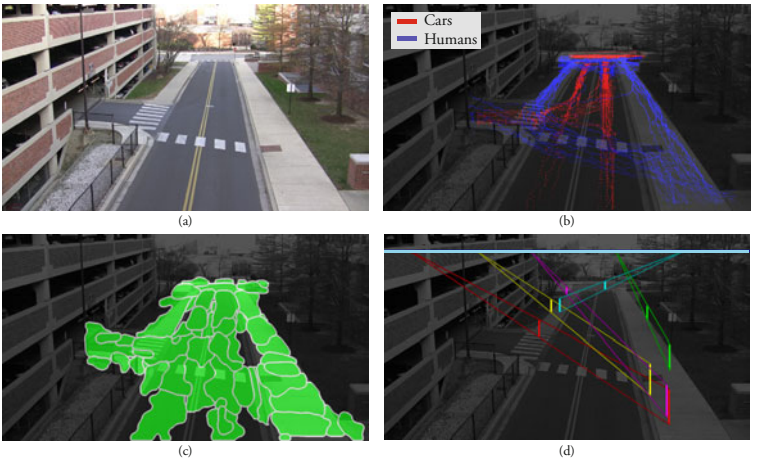


Fig. 3. Components of the image analysis module. (a) Background image for Scene I. (b) Trajectories (Sec 3.1). (c) Zones (Sec 3.2). (d) Horizon line estimate (Sec 3.3).

3.2 Zone Segmentation

The MLN based reasoning module utilizes events generated by the image analysis framework to assign labels to each part of the scene. To avoid performing inference at the pixel level, we segment the scene spatially into a set of zones, and perform inference on each zone. Zone segmentation groups pixels based on their appearance, location and the motion characteristics of objects passing through them. This results in a set of zones in which objects display distinct behaviors. Examples include locations where people gather and stand still for a long time (at bus stops), locations where vehicles drive in specific directions (along drive lanes), locations where cars and people cross each other (at cross walks), etc.

[1] Code obtained: `http://www.umiacs.umd.edu/~schwartz/softwares.html`

We begin by obtaining a background image by simply constructing an image for which a pixel $p(i, j)$ is the median of all pixels in the video at that location. This image is oversegmented by an image segmentation algorithm [27] to create a set of superpixels[2]. A set of features are computed for each superpixel, including: (1) Appearance - 3 histograms (one each for R,G,B) (2) Motion - Velocity magnitude histogram and velocity orientation histograms (weighted by magnitude) for each class of passing objects. An affinity matrix that includes the similarity between all pairs of superpixels is created for each feature. The distance metric used for all histograms is the Earth Mover's Distance (EMD). In addition, a location based affinity matrix is also created. This captures the minimum Euclidean distance between all pairs of superpixels and is calculated efficiently using the distance transform. Spectral clustering is then used to group superpixels into zones. We used the self-tuning method proposed by Zelnik-Manor et al. [28][3], since it automatically selects the scale of analysis as well as the number of clusters. Figure 3c shows zones obtained for one of the scenes in our dataset.

3.3 Scene Geometry Analysis

Surface Layout. An estimate of the scene surface layout supports reasoning about the location of many scene elements. For example, entrance and exit zones (such as doors into buildings) are typically located where horizontal and vertical surfaces meet. We obtain a rough surface layout using the method of [5][4] which classifies pixels into three primary classes: *horizontal*, *vertical* and *sky*. This estimate uses information extracted from individual images. However, we also have the additional knowledge of object trajectories that can help us obtain better surface estimates. Our meta-rules (discussed in Section 4) encode common sense knowledge about surfaces such as: *Objects are supported by a horizontal surface. Objects might appear out of and disappear into vertical surfaces.* Such rules allow us to correct some of the erroneous surface estimates provided by [5]. Figure 4 shows a surface layout before and after inference by our system.

Proximity Measures. Models of scene elements typically contain predicates corresponding to notions of proximity in the world, such as *nearby*, *far away*, *next to*, etc. Distances measured directly in the image plane, however, do not maintain these scene proximity relationships. Under a unit aspect ratio perspective camera model, we show how to compare segment lengths measured at different parts of the image based on their *true lengths* in the 3D world. We break the problem down into two components: segments parallel to the camera axis (lengths along a column of pixels) and segments parallel to the camera image plane (lengths along a row of pixels), shown in Figure 5.

[2] Code obtained: `http://www.wisdom.weizmann.ac.il/~ronen/index_files/segmentation.html`

[3] Code obtained: `http://www.vision.caltech.edu/lihi/Demos/SelfTuningClustering.html`

[4] Code obtained: `http://www.cs.uiuc.edu/homes/dhoiem/`

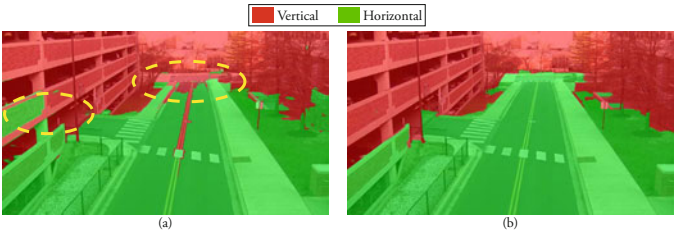


Fig. 4. Surface layout estimates before and after inference by our system. The road visible in the far distance is erroneously labeled as a vertical surface (in (a)), but corrected after inference (in (b)), due to the presence of objects passing over it.

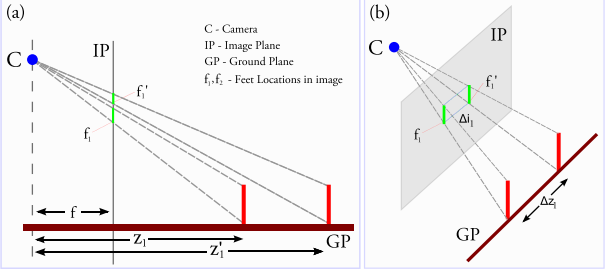


Fig. 5. Schematic relating image plane distances to ground plane distances

Consider Figure 5a. As in [6], we translate our image co-ordinates (u, v) to $(\hat{u}, \hat{v})$ so that $\hat{v} = 0$ for every point on the horizon line and $\hat{v} > 0$ below the horizon line. In this new co-ordinate system f_1 represents the foot location in the image of a person at a distance z_1 from the camera and f_1' is the foot location when the person takes a step Δz_1 parallel to the camera axis to be located at a distance z_1' from the camera. Now, $f_1 z_1 = f_1' z_1' = f y_c$. Consider a person at a second location in the scene taking a step Δz_2. This gives us: $f_2 z_2 = f_2' z_2' = f y_c$. A little algebra yields $\frac{(f_1' - f_1) f_2 f_2'}{(f_2' - f_2) f_1 f_1'} = \frac{\Delta z_1}{\Delta z_2}$. Now consider Figure 5b. Here the person moves from foot location f_1 to a new location f_1' parallel to the camera image plane. One can obtain: $\Delta i_1 y_c = \Delta z_1 f_1$, where Δi_1 represents the image plane distance between the two feet locations. For a second person at a new location, we obtain: $\Delta i_2 y_c = \Delta z_2 f_2$. This yields $\frac{\Delta i_1}{\Delta i_2} \frac{f_2}{f_1} = \frac{\Delta z_1}{\Delta z_2}$. Given the horizon line, the above equations relate distances (segment lengths) measured at different locations in the image plane, based on the true 3D measurements. Measures such as *nearby*, *far away*, etc., when defined at one location in the image, can be thus transformed to equivalent measures at other locations.

The horizon line is estimated using the method of Lv et al. [29]. Consider two vertical poles of the same height in the scene. The two lines joining their foot locations and head locations, respectively, intersect at a point on the horizon line. Thus, three non-coplanar poles of the same height uniquely determine the

horizon line. In practice, we have a large number of people walking through each scene. Each pair of detections (from the same human track) provides us with an estimate of a point lying on the horizon line. A least squares estimate of many such detection pairs yields a good horizon line estimate (shown in Figure 3d).

Zone Transitions. While the distance measures described above help define notions of proximity in the scene, they do not capture the restrictions imposed on object trajectories due to the scene layout. For example, a sidewalk is located adjacent to a road, yet vehicles typically do not traverse between roads and sidewalks. We characterize typical traffic patterns in the scene in terms of the average transition times of objects between one zone and another. These patterns are represented as transition matrices, one for each object class. Zone pairs that do not have any traffic flowing between them, are assigned a large transition time by default. Figure 6 shows examples of proximal zones. Note that cars typically conform to fixed directions, where as people walk along paths in both directions.

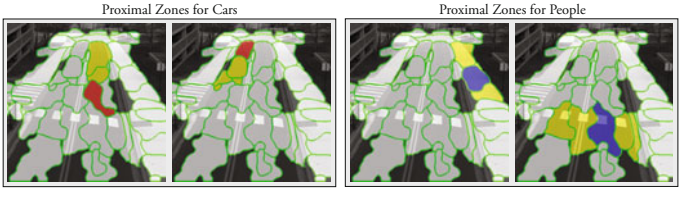


Fig. 6. Examples of proximal zones based on zone transition matrices. (a) Vehicles travel from red zones onto yellow zones within a short time span. (b) People walk from blue zones onto yellow within a short time span.

Directionality. User descriptions of scene elements often involve spatial prepositions which provide a notion of directionality, such as *in front of*, *behind*, *to the left of*, etc. Under the assumption that objects move in the direction in which they are facing, we define four directions with respect to the motion of the object: left, right, front and behind. Furthermore, some zones in the scene exhibit a single dominant direction of motion (based on the objects that pass through them). This is especially true of zones located on the road, on which vehicles strictly follow a single direction of motion. The four directions defined above are also noted for such zones, with respect to the centroid of the given zone.

3.4 Zone Occlusion Relationships

As objects move through the scene, they occlude different areas of the scene as well as objects present at those locations. This is a common source of errors in a typical computer vision system. Knowledge about typical occlusion areas can provide valuable information to the scene understanding framework. For example, people trajectories ending at a location suggest the presence of a doorway to a building at that location. However, the observation of a vehicle parked nearby,

with the knowledge that it may causes occlusions at the former location, can prevent such an inference error. We represent occlusion relationships between zones using a binary matrix OC (one for each object class). For every object that passes through a zone z_i, we determine zones in the scene that intersect the object bounding box in the image plane (indicating potential occlusions), while the object was within z_i. If a zone z_j consistently undergoes occlusion by objects in z_i, the indicator variable $OC(z_i, z_j)$ is set to 1.

3.5 Event Generation

Short time spans of 20 frames are grouped together to form a temporal window. A set of dynamic events is generated at every zone within each temporal window. These events characterize the location, motion and trajectory of objects in a given zone during the given window. This results in a large set of evidence ground atoms passed to the inference module throughout the duration of the video sequence. In addition, the image analysis module also generates a set of zone characteristics and inter zone relationships, as described above. These are also represented as evidence atoms and passed on to the inference module.

4 Knowledge Base

The knowledge base consists of two components: a set of scene element models and a set of meta-rules that capture information about scene geometry, occlusion reasoning as well as common sense knowledge that applies to many domains. We begin with a description of our approach to represent uncertain knowledge, and then proceed with outlining the two components of our knowledge base.

4.1 Knowledge Representation

Knowledge is represented as first order production rules. The rules are represented in clausal form, whereby each rule is a conjunction of clauses and each clause is a disjunction of literals. Rules are constructed using variables such as *zone*, *time*, etc. Some variables are typed. Such variables have mutually exclusive and exhaustive values. For example, the typed variable *appearPersonReason* signifies an explanation for the birth of a person track and must take one of the following values: {*TrackingFailure, OcclusionByCar,...*}.

We use two types of predicates. The first represents events in the video and are associated with a particular zone and time instant (*PersonAppear(zone,time)*). The second represents properties of individual zones (*ZoneIsVertical(zone)*), relationships between zones (*ZoneNearZone(zone, zone)*) and relationships between time instants (*ShortlyAfter(time, time)*). These predicates need only be calculated once for the entire video sequence.

Each rule in our knowledge base is associated with a weight that indicates its confidence. We use three degree of confidence for rules of absolute certainty ($weight = M$), for ones with lesser certainty ($0.5M$) and for rules that may be

true a very small fraction of times ($0.25M$). One may infer the certainty of a human elicited rule by frequency adverbs such as always, never, rarely, etc.

Some of the predicates generated by the image analysis module, such as *ZoneIsVertical(zone)*, have a confidence value associated with them. Such uncertain predicates are integrated into the first order rules using the method employed in [8]. Consider a predicate P with a weight w. We introduce a dummy observation predicate O_P along with a rule $O_P \rightarrow P$ and associate the weight w with this rule. The predicate O_P does not have any weight associated with it.

4.2 Scene Element Models

Each scene element is described by a logical model comprising a set of first order rules. These logical models describe a scene element on the basis of *what typically happens* in a scene at that element. For example, the logical model for a crosswalk consisting of logic rules with confidence measures is given in Figure 7[5]. The numbers in parentheses represent the weight assigned to each rule (recall that M represents the highest weight assigned in the knowledge base). The presence of people walking on the road indicates that they might be passing over a crosswalk (Rule 1). However, pedestrians often disobey laws and cross the road at other locations. The presence of a car waiting for people to cross the road is a stronger indication of a crosswalk and is thus assigned a higher weight (Rule 2).

Crosswalk Model:

```
Rule1: (0.25M)  PeopleMove(z1,t1) ^ ZoneClassA(z1,Road) => ZoneClass(z1,Crosswalk)
Rule2: (0.5M)   PeopleMove(z1,t1) ^ ZoneClassA(z1,Road) ^ CarStop(z2,t1) ^
                ZoneTransitionCar(z2,z1) => ZoneClass(z1,Crosswalk)
Rule3: (0.5M)   ZoneClassA(z1,Road) ^ ZoneTransitionPeople(z2,z1) ^ ZoneClassA(z2,Sidewalk) ^
                ZoneTransitionPeople(z1,z3) ^ ZoneClassA(z3,Sidewalk) => ZoneClass(z1,Crosswalk)
Rule4: (1.0M)   !ZoneClass(z1,Road) => !ZoneClass(z1,Crosswalk)
```

Fig. 7. First order logic rules representing a crosswalk model

4.3 Meta-Rules

In addition to the scene element models, the knowledge base also consists of a set of meta-rules, which encode information relating to scene geometry, occlusion handling, common failures of low level computer vision modules as well as common sense knowledge about the world. They only need to be written once, but are then widely applicable over a large number of domains. For instance, consider the scene element *Building Entrance/Exit*. Entrances and exits are typically characterized as sources and sinks of person tracks. There are however, a variety of situations that may lead to an initiation of a person track such as: exiting a vehicle, tracker identity switching, occlusion within a group of people, etc. Our meta rules encode such possibilities. This enables the inference module to reason about plausible explanations when it encounters a new person track. This reduces the number of false locations that might be labeled as an entrance-exit.

[5] Other models provided at: http://www.umiacs.umd.edu/~ani

5 Inference Using Markov Logic Networks

There has been a growing interest in problems related to knowledge representation and learning in domains that are rich in relational as well as probabilistic structure. Markov Logic Networks (MLN) are one such representation that combine first order logic with probability theory in finite domains [30]. They support the specification of statistical models using intuitive and understandable first order rules. A first order knowledge base, by itself, is often impractical to use for real world problems. Each rule in such a knowledge base is a hard constraint. A world that does not satisfy a single formula gets assigned a zero probability. MLNs attempt to relax these hard constraints using weights for each formula. The probability of a world is dependent upon the number of formulae that the world satisfies and the weights assigned to those formulae. MLNs can also be viewed as a template for constructing ordinary Markov networks. Given a set of formulae and constants, a MLN produces a Markov network. Based on the constructed network, marginal distributions of events given the observations can be computed using probabilistic inference. We use the Alchemy system [31] to represent our rules and perform inference on the resulting MLN[6].

5.1 Local Inference Procedures

The image analysis module generates a large number of evidence ground atoms within every temporal window, for every zone in the scene. Over the entire video, the number of ground atoms gets prohibitively large, rendering inference intractable. However, the spatio temporal interactions between objects, that characterize the scene elements of interest are sufficiently local in nature, both spatially and temporally. For instance, consider the crosswalk model in Figure 7 described by the interaction between people walking on the crosswalk and vehicles waiting on the road adjacent to it. Interactions between objects at locations far away from the crosswalk do not affect inference about the given zone. Likewise, interactions between people and vehicles at the crosswalk, at other times in the video, are largely independent of the current interaction.

We break down the large inference problem into smaller ones, carried out in every zone and at regularly spaced time instants. For every such spatio temporal location, the inference procedure takes into consideration events generated at a set of neighboring zones and time instants. For each zone, votes for each label, which are generated over the duration of the video, are aggregated to determine the final scene element label associated with that zone.

6 Experiments

We demonstrate our scene understanding framework on a dataset of 5 videos of public spaces, totaling over 100,000 frames (about 58 minutes). The video data

[6] Code available: http://alchemy.cs.washington.edu/

has been collected using cameras overlooking scenes from varying viewpoints. Each scene contains a large amount of pedestrian, car and bus traffic passing through it. Over the entire dataset, the number of pedestrians, cars and buses is approximately 700, 500 and 25 respectively. The data has been collected in high definition mode (1920x1080 pixels). Figure 8 shows some representative frames.

The scene elements that we seek to identify are: Road, Sidewalk, Other Path (other paths taken by people, which are not sidewalks), Bus-stops, Stop-sign Zones, Crosswalks, Entrances-Exits for People (typically buildings) and Entrances-Exits for Vehicles (typically garages). Figure 8 shows the labels assigned to different regions of the scenes. The system is able to correctly identify a large number of the scene elements using the human elicited domain models.

Our scene understanding framework is effectively able to reason about the scene geometry and occlusions to identify scene elements from widely varying viewpoints. Recall the example of a bus-stop observed from two viewpoints (Figure 2). Scene

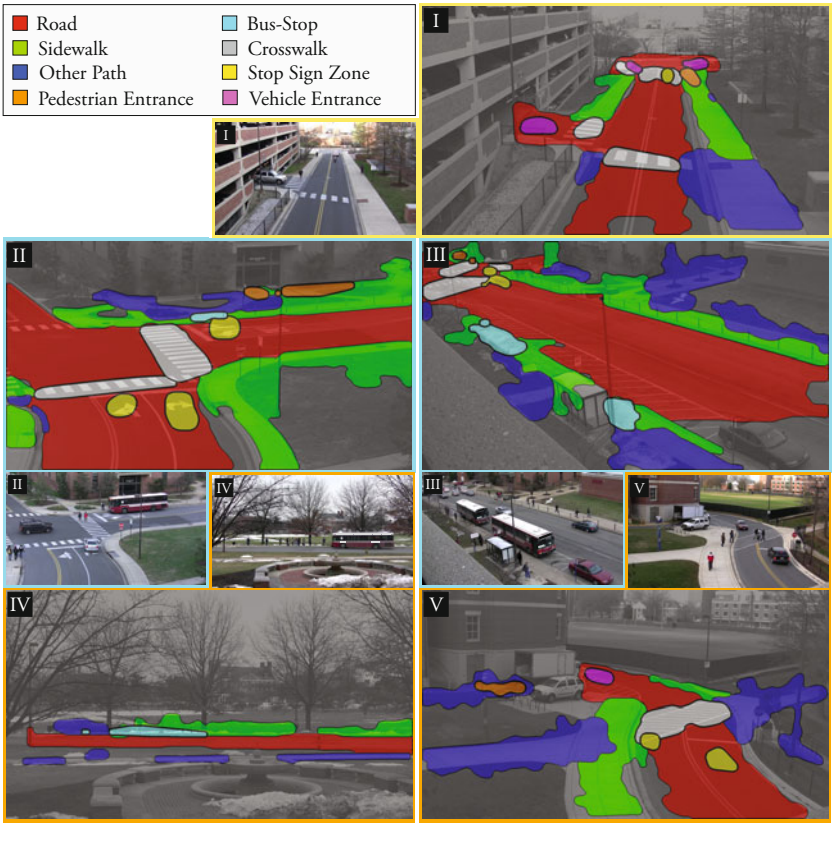


Fig. 8. Scene element labels determined by our system for Scenes I-V along with a representative image from each scene

III contains a view of a bus-stop in which we are able to observe people entering and exiting the bus. Scene II and IV, on the other hand, contain views of bus-stops in which the doors of the bus are not visible. The system reasons about people that might have entered and exited the buses that stopped at the location and correctly identifies all bus stops. Two locations are marked as bus-stops in Scene III, since buses stop one behind the other in this scene.

Pedestrian crosswalks are also correctly identified in all scenes, with the exception of a partially visible crosswalk in Scene II. These include the three crosswalks visible in the far distance in Scene III. A fair number of people tend to cross roads at locations other than crosswalks. However, cars do not always stop for such jaywalking violations. The system correctly identifies crosswalk locations using this additional information and suppresses the false alarms. Vehicle and pedestrian entrances are identified on the basis of track appearances and disappearances into vertical surfaces. Scene I shows a correctly identified garage entrance. The other detections in Scene I are not garage entrances, but they correspond to locations in the scene (away from the image boundary and close to vertical surfaces) where cars enter and exit the camera frame. Scene V shows a loading dock correctly marked as an entrance/exit for people. We fail to detect one of the doorways in Scene III (primarily due to a leafless, yet occluding tree), but another entrance in the distance away is correctly determined.

Roads, Sidewalks and Other Paths are also identified in each scene. Sidewalks are defined to be paths adjacent to roads and parallel to them on which people walk. Zones are considered parallel to one another if the orientations of objects passing through them are similar. Stop-sign zones are also detected in the scenes. The system does not merely depend on locations where cars stop-and-go, but also uses information such as *Stop zones are located adjacent to cross-walks and at intersections*. Scene V shows a false alarm caused by cars frequently stopping at a busy crosswalk. Such false alarms can be reduced by analyzing a larger amount of data, spanning different times of the day.

Acknowledgements. This research was partially supported by ONR grant N00014-09-10044.

References

1. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
2. Schwartz, W., Kembhavi, A., Harwood, D., Davis, L.: Human detection using partial least squares analysis. In: ICCV (2009)
3. Lampert, C., Blaschko, M., Hofmann, T.: Efficient subwindow search: A branch and bound framework for object localization. IEEE PAMI (2009)
4. Huang, C., Wu, B., Nevatia, R.: Robust object tracking by hierarchical association of detection responses. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 788–801. Springer, Heidelberg (2008)
5. Hoiem, D., Efros, A.A., Hebert, M.: Recovering surface layout from an image. IJCV (2007)
6. Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. In: CVPR (2006)

7. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions (2009)
8. Tran, S.D., Davis, L.S.: Event Modeling and Recognition Using MLNs. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 610–623. Springer, Heidelberg (2008)
9. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. IJCV (2001)
10. Fei-fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: CVPR (2005)
11. Bosch, A., Zisserman, A., Munoz, X.: Scene classification using a hybrid generative/discriminative approach. IEEE PAMI (2008)
12. Gupta, A., Kembhavi, A., Davis, L.: Observing human-object interactions: Using spatial and functional compatibility for recognition. IEEE PAMI (2009)
13. Bosch, A., Zisserman, A., Muñoz, X.: Scene classification via plsa. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 517–530. Springer, Heidelberg (2006)
14. 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)
15. Li, L.-J., Socher, R., Fei-Fei, L.: Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In: CVPR (2009)
16. Charless, X.R., Ren, X., Fowlkes, C.C., Malik, J.: Figure/ground assignment in natural images. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 614–627. Springer, Heidelberg (2006)
17. Hoiem, D., Efros, A.A., Hebert, M.: Closing the loop on scene interpretation. In: CVPR (2008)
18. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. IEEE PAMI (2000)
19. Makris, D., Ellis, T.: Learning semantic scene models from observing activity in visual surveillance. IEEE Trans. Systems, Man, and Cybernetics (2005)
20. Hu, W., Xiao, X., Fu, Z., Xie, D., Tan, T., Maybank, S.: A system for learning statistical motion patterns. IEEE PAMI (2006)
21. Saleemi, I., Shafique, K., Shah, M.: Probabilistic modeling of scene dynamics for applications in visual surveillance. IEEE PAMI (2009)
22. Swears, E., Hoogs, A.: Functional scene element recognition for video scene analysis. In: Workshop on Motion and Video Computing (2009)
23. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer(2009)
24. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: CVPR (2009)
25. Wang, J., Markert, K., Everingham, M.: Learning models for object recognition from natural language descriptions (2009)
26. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. IJCV (2008)
27. Alpert, S., Galun, M., Basri, R., Brandt, A.: Image segmentation by probabilistic bottom-up aggregation and cue integration. In: CVPR (2007)
28. Zelnik-manor, L., Perona, P.: Self-tuning spectral clustering. In: NIPS (2004)
29. Lv, F., Zhao, T., Nevatia, R.: Camera calibration from video of a walking human. IEEE PAMI (2006)
30. Richardson, M., Domingos, P.: Markov logic networks. Machine Learning (2006)
31. Kok, S., Sumner, M., Richardson, M., Singla, P., Poon, H., Lowd, D., Domingos, P.: The alchemy system for statistical relational ai. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA (2007)

A Data-Driven Approach for Event Prediction

Jenny Yuen and Antonio Torralba

CSAIL MIT
{jenny,torralba}@csail.mit.edu

Abstract. When given a single static picture, humans can not only interpret the instantaneous content captured by the image, but also they are able to infer the chain of dynamic events that are likely to happen in the near future. Similarly, when a human observes a short video, it is easy to decide if the event taking place in the video is normal or unexpected, even if the video depicts a an unfamiliar place for the viewer. This is in contrast with work in surveillance and outlier event detection, where the models rely on thousands of hours of video recorded at a single place in order to identify what constitutes an unusual event. In this work we present a simple method to identify videos with unusual events in a large collection of short video clips. The algorithm is inspired by recent approaches in computer vision that rely on large databases. In this work we show how, relying on large collections of videos, we can retrieve other videos similar to the query to build a simple model of the distribution of expected motions for the query. Consequently, the model can evaluate how unusual is the video as well as make event predictions. We show how a very simple retrieval model is able to provide reliable results.

1 Introduction

If we are told to visualize a street scene, we can imagine some composition with basic elements in it. Moreover, if we are asked to imagine what can happen in it, we might say there is a car moving through a road, being in contact to the ground and preserving some velocity and size relationships with respect to other elements in the scene (say a person or a building). Even when constrained by its

Fig. 1. What do these images have in common? They depict objects moving towards the right. These images do not contain motion cues such as temporal information or motion blur. The implied motion is known because we can recognize the image content and make reliable predictions what would occur if these were movies playing.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 707–720, 2010.

composition (e.g. when being shown a picture of it) we can predict things like an approximate speed of the car, and maybe even its direction (see fig. 1). Human capacity for mental imagery and story telling is driven by the years of prior knowledge we have about our surroundings. Moreover, it has been found that static images implying motion are also important in visual perception and are able to produce motion after-effects [1] and even activate motion sensitive areas in the human brain [2]. As a consequence, the human visual system is capable of accurately predicting plausible events in a static scene (or future events in a video sequence) as well as is finely tuned to flag unusual configurations or events.

Event and action detection are well-studied topics in computer vision. Several works have proposed models to study, characterize, and classify human actions ranging from constrained environments [3,4] to actions in the wild such as TV shows, sporting events, and cluttered backgrounds [5,6]. In this scenario, the objective is to identify the action class of a previously unknown query video given a training dataset of action exemplars (captured at different locations). A different line of work is that of event detection for video surveillance applications. In this case, the algorithm is given a large corpus of training video captured at a particular location as input, and the objective is to identify abnormal events taking place in the future in that same scene [7,8,9,10]. Consequently, deploying a surveillance system requires days of data acquisition from the target and hours of training for each new location.

In this paper we look into the problem of generic event prediction for scene instances different from the ones in some large training corpus. In other words, given an image (or a short video clip), we want to identify the possible events that may occur as well as the abnormal ones. We motivate our problem with a parallel to object recognition. Event prediction and anomaly detection technologies for surveillance are now analogous to object instance recognition. Many works in object recognition are moving towards the more generic problem of object category recognition [11,12]. We aim to push the envelope in the video aspect by introducing a framework that can easily adapt to new scene instances without the requirement of retraining a model for each new location. Moreover, other potential applications lie in the areas of video collection retrieval in online services such as YouTube, Vimeo, where video clips are captured in different locations and greatly differ with respect to controlled video sources such as surveillance feeds and tv programming as was pointed out by Zanetti *et al.* [13].

Given a query image, our purpose is to identify the events that are likely to take place in it. We have a rich video corpus with 2401 real world videos acting as our prior knowledge of the world. In an offline stage, we generate and cluster motion tracks for each video in the corpus. Using scene-matching, our system retrieves videos with similar image content. Track information from the retrieved videos is integrated to make a prediction of where in the image motion is likely to take place. Alternatively, if the input is a video, we track and cluster salient features in the query and compare each to the ones in the retrieved neighbor set. A track cluster can then be flagged as unusual if it does not match any in the retrieved set.

2 Related Work

Human action recognition is a popular problem in the video domain. The work by Efros *et al.* [14] learns optical flow correlations of human actions in low resolution video. Schechtman and Irani exploit self similarity correlations in space-time volumes to find similar actions given an exemplar query. Niebles *et al.* [5] characterize and detect human actions under complex video sequences by learning probability distributions of sparse space-time interest points. Laptev *et al.* densely extracts spatio-temporal features in a grid and uses a bag of features approach to detect actions in movies. Messing *et al.* models human activities as mixtures of bags of velocity trajectories, extracted from track data. None of these works study the task of event prediction and are constrained to human actions. Similar in concept to our vision is the work by Li *et al.* [15], where the objective is action classification given an object and a scene . Our work is geared towards localized prediction including trajectory generation, not classification.

Extensive work has also taken place in event and anomaly detection for surveillance applications. A family of works relies on detecting, tracking, and classifying objects of interest and learning features to distinguish events. Dalley *et al.* detect loitering and bag dropping events using a blob tracker to extract moving objects and detect humans and bags. The system idenfifies a loitering event if a person blob does not move for a period of time. Bag dropping events are detected by checking the distance between a bag and a person; if the distance becomes larger than some threshold, it is identified as a dropped bag. A second family of works clusters motion features and learning distributions on motion vectors across time. Wang *et al.* [7] uses a non-parametric Bayesian model for trajectory clustering and analysis. A marginal likelihood is computed for each video clip, and low likelihood events are flagged as abnormal. One common assumption of these methods is that training data for each scene instance where the system will be deployed is available. Therefore, the knowledge built is not transferrable to new locations, as the algorithm needs to be retrained with video feeds from each new location to be deployed.

Numerous works have demonstrated success using a rich databases for retrieving and/or transferring information to queries in both image [16,17,18,19] and video [20,21]. In video applications, Sivic *et al.* [21], proposed a video representation for exemplar-based retrieval within the same movie. Moving objects are tracked and their trajectories grouped. Upon selection of an image crop in some video frame, the system searches across video key frames for similar image regions and retrieves portions of the movie containing the queried object instance. The work proposed by Liu *et al.* [20] is the closest one to our system. It introduces a method for motion synthesis from static images by matching a query image to a database of video clip frames and transferring the moving regions from the nearest neighbor videos (identified as regions where the optical flow magnitude is nonzero) to the static query image. This work constructs independent interpretations per nearest neighbors. Instead, our work builds localized motion maps as probability distributions after merging votes from several nearest neighbors. Moreover, we aim to have a higher level representation where each moving object is modeled as a track blob

while [20] generates hypotheses as one motion region per frame. In summary, these works demonstrate the strong potential of data-driven techniques, which to our knowledge no prior work has extended into anomaly detection.

3 Scene-Based Video Retrieval

The objective of this project is to use event knowledge from a training database of videos to construct an event prediction for a given a static query image. To achieve some semantic coherence, we want to transfer event information only from similar images. Therefore, we need a good retrieval system that will return matches with similar scene structures (*e.g.* a picture of an alley will be matched with another alley photo shot with a similar viewpoint) even if the scene instances are different. In this paper we will explore the usage of two scene matching techniques: GIST [22] and spatial pyramid dense SIFT [23] matching. The GIST descriptor encodes perceptual dimensions that characterize the dominant spatial structure of a scene. The spatial pyramid SIFT matching technique works by partitioning an image into subregions and computing histograms of local features at each sub-region. As a result, images with similar global geometric correspondence can be easily retrieved. The advantage of both the GIST and dense SIFT retrieval methods is their speed and efficiency at projecting images into a space where similar semantic scenes are close together. This idea has proven robust in many non-parametric data-driven techniques such as label transfer [17] and scene completion [18] amongst many others. To retrieve nearest videos from a database, we perform matching between the first frame of the video query and the first frame of each of the videos in the database.

4 Video Event Representation

We introduce a system that models a video as a set of trajectories of keypoints throughout time. Individual tracks are further clustered into groups with similar motion. These clusters will be used to represent events in the video.

4.1 Recovering Trajectories

For each video, we extract trajectories of points in the sequence using an implementation of the KLT tracker [24] by Birchfield [25]. The KLT tracking equation seeks the displacement $\mathbf{d} = [d_x, d_y]^T$ that minimizes the dissimilarity amongst two windows, given a point $\mathbf{p} = [x, y]^T$ and two consecutive frames I and J:

$$\varepsilon(w) = \int\int_W [J(\mathbf{p} + \frac{\mathbf{d}}{2}) - I(\mathbf{p} - \frac{\mathbf{d}}{2})]^2 w(\mathbf{p}) d\mathbf{p} \tag{1}$$

where W is the window neighborhood, and $w(\mathbf{d})$ is the weighing function (set to 1). Using a Taylor series expansion of J and I, the displacement that minimizes ε is:

$$\frac{\partial \varepsilon}{\partial \mathbf{d}} = \int \int_W [J(\mathbf{p}) - I(\mathbf{p}) + \mathbf{g}^T(\mathbf{p})\mathbf{d}]\mathbf{g}(\mathbf{p})w(\mathbf{p})d\mathbf{p} = 0 \quad (2)$$

where $\mathbf{g} = \left[\frac{\partial}{\partial x}\left(\frac{I+J}{2}\right) \frac{\partial}{\partial y}\left(\frac{I+J}{2}\right)\right]^T$

The tracker finds salient points by examining the minimum eigenvalue of each 2 by 2 gradient matrix. We initialize the tracker by extracting 2000 salient points at the first video frame. The tracker finds the correspondences of the points sequentially throughout the frames in the video. Whenever a track is broken (a point is lost due to high error or occlusions), new salient points are detected to maintain a consistent number of tracks throughout the video. As a result, the algorithm produces tracks, which are sequences of location tuples $\mathbf{T} = (x(t), y(t))_{t \in \mathbf{D}}$ within a duration $\mathbf{D}$ for each tracked point. For more details on the implementation, we refer to the the original KLT tracker paper.

4.2 Clustering Trajectories

Now that we have a set of trajectories for salient points in an image, we proceed to group them at a higher level. Ideally, tracks from the same object should be clustered together. We define the following distance function between two tracks

$$d_{track}(\mathbf{T}_i, \mathbf{T}_j) \equiv \frac{1}{|\mathbf{D}_i \cap \mathbf{D}_j|} \sum_{t \in \mathbf{D}_i \cap \mathbf{D}_j} \sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2} \quad (3)$$

We use the distance function to create an affinity matrix between tracks and use normalized cuts [26] to cluster them. Each entry of the affinity matrix is defined as $\mathbf{W}_{ij} = \exp(-d_{track}(\mathbf{T}_i, \mathbf{T}_j)/\sigma^2)$. The clustering output will thus be a group label assignment to each track. See fig. 3 for a visualization of the data. Since we do not know the number of clusters for each video in advance, we set a value of 10. In some cases this will cause an over segmentation of the tracks and will generate more than one cluster for some objects.

4.3 Comparing Track Clusters

For each track cluster $\mathbf{C} = \{\mathbf{T}_i\}$, we quantize the instantaneous velocity of each track point into 8 orientations To ensure rough spatial coherency between clusters, we superimpose a regular grid with a cell spacing of 10 pixels on top of the image frame to create a spatial histogram containing 8 sub-bins at each cell in the grid. Let H_1 and H_2 denote the histograms formed by the track clusters $\mathbf{C}_1$ and $\mathbf{C}_2$ such that $H_1(i, b)$ and $H_2(i, b)$ denote the number of velocity points from the first and second track clusters respectively that fall into the bth sub-bin of the ith bin of the histogram, where $i \in \mathbf{G}$ and $\mathbf{G}$ denotes the bins in the grid. We define the similarity between two track clusters as the intersection of their velocity histograms:

$$\mathbf{S}_{clust}(\mathbf{C}_1, \mathbf{C}_2) \equiv \mathbf{I}(H_1, H_2) = \sum_{i \in \mathbf{G}} \sum_{b=1}^{8} min\Big(H_1(i, b), H_2(i, b)\Big) \quad (4)$$

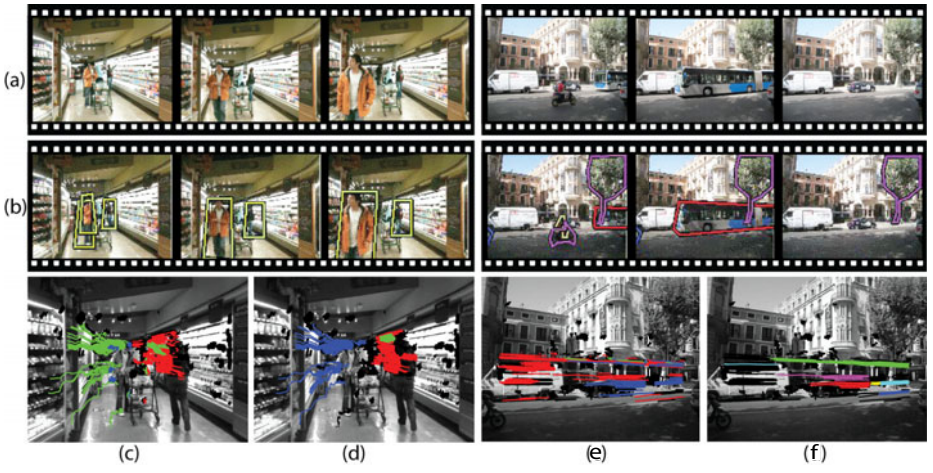


Fig. 2. Track clustering. Sample frames from the video sequence (a). The ground truth annotations denoted by polygons surrounding moving objects (b) can be used to generate ground truth labels for the tracked points in the video (c). Our track distance affinity function is used to to automatically cluster tracks into groups and generates fairly reasonable clusters where each roughly correspond to independent objects in the scene (d). The track clusters visualizations in (c) and (d) show the first frame of each video and the spatial location of all tracked points for the duration of the clip color-coded by the track cluster that each point corresponds to.

This metric was designed in the same spirit as the bottom level of the spatial pyramid matching method by Lazebnik *et al.* . We aim for matches that approximately preserve global spatial correspondences. Since our video neighbor knowledge-base is assumed to be spatially aligned to our query, a good match shall also preserve an approximate similar spatial coherence.

5 Video Database and Ground Truth

Our database consists of 2277 videos belonging to 100 scene categories. The categories with the most videos are: street (809), plaza (135), interior of a church (103), crosswalk (82), and aquarium (75). Additionally, 14 videos containing unusual events were downloaded from the web (see fig. 3 for some sample frames). 500 of the videos originate from the LabelMe video dataset [27]. As these videos were collected using consumer cameras without a tripod, there is slight camera shake. Using the LabelMe video system, the videos were stabilized. The object-level ground truth labeling in the LabelMe video database allows us to easily visualize the ground truth clustering of tracks and compare it with our automated results (see fig. 2). We split the database into 2301 training videos, selected 134 fully videos from outdoor urban scenes and the 14 unusual videos to create a test set with 148 videos.

Fig. 3. Unusual videos. We define an unusual or anomalous event as one that is not likely to happen in our training data set. However, we ensured that they belong to scene classes present in our video corpus.

6 Experiments and Applications

We present two applications of our framework. Given the information from nearest neighbor videos, what can we say about the image if we were to see it in action? As an example, we can make good predictions of where motion is bound to happen in an image. We also present a method for determining the degree of anomaly of an event in a video clip using our training data.

6.1 Localized Motion Prediction

Given a static image, we can generate a probabilistic map determining the spatial extent of the motion. In order to estimate $p(motion|x, y, \text{scene})$ we use a parzen window estimator and the trajectories of the N=50 nearest neighbor videos retrieved with scene matching methods (GIST or dense SIFT-based).

$$p(motion|x, y, \text{scene}) = \frac{1}{N}\sum_{i}^{N}\frac{1}{M_i}\sum_{j}^{M_i}\sum_{t\in D} K(x - x_{i,j}(t), y - y_{i,j}(t); \sigma) \quad (5)$$

where N is the number of videos and M_i is the number of tracks in the ith video and $K(x, y; \sigma)$ is a gaussian kernel of width σ^2. Fig. 4 a shows the per-pixel prediction ROC curve compared using gist nearest neighbors, dense SIFT matching, and as a baseline, a random set of nearest neighbors. The evaluation set is composed of the first frame of each test video. We use the location of the tracked points in the test set as ground truth. Notice that scenes can have multiple plausible motions occurring in them but our current ground truth only provides one explanation. Despite our limited capacity of evaluation, notice the improvement when using SIFT and GIST matching to retrieve nearest neighbors. This graph suggests that (1) different sets of motions happen in different scenes, and (2) scene matching techniques do help filtering out distracting scenes to

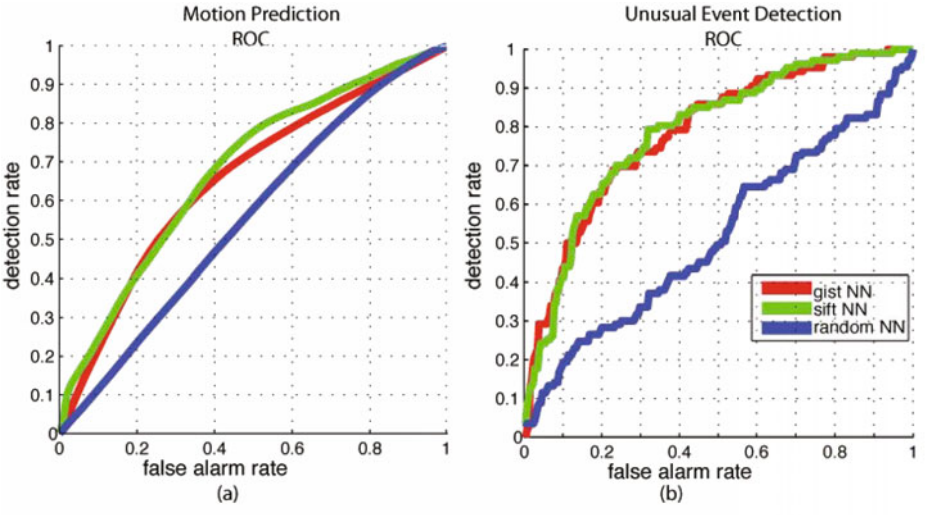


Fig. 4. Localized motion prediction (a) and unusual event detection (b). The algorithm was compared against two scene matching methods (GIST and dense SIFT) as well as a baseline supported by random nearest neighbors. Retrieving videos similar to the query image improves the classification rate.

make more reliable predictions (for example, a person climbing the wall of a building in a street scene would be considered unusual but a person climbing a wall in a rock climbing scene is normal). Fig. 6 c and 7 c contain the probability motion map constructed after integrating the track information from the nearest neighbors of each query video depicted in column (a). Notice that the location of high probability regions varies depending on the type of scenes. Moreover, the reliability of the motion maps depends on (1) how accurately the scene retrieval system returns nearest neighbors from the same scene category (2) whether the video corpus contains similar scenes. The reader can get an intuition of this by looking at column (e), which contains the average nearest neighbor image.

6.2 Event Prediction from a Single Image

Given a static image, we demonstrated that we can generate a probabilistic function per pixel. However, we are not only constrained to per-pixel information. We can use the track clusters of videos retrieved from the database and generate coherent track cluster predictions. One method is by directly transferring track clusters from nearest neighbors into the query image. However, this might generate too many similar predictions. Another way lies in clustering the retrieved track clusters. We use normalized cuts clustering for this step at the track cluster level using the distance function described in equation 4 to compare pairs of track clusters. Fig. 5 shows example track clusters overlaid on top of the static query image. A required input to the normalized cuts algorithm is the number of

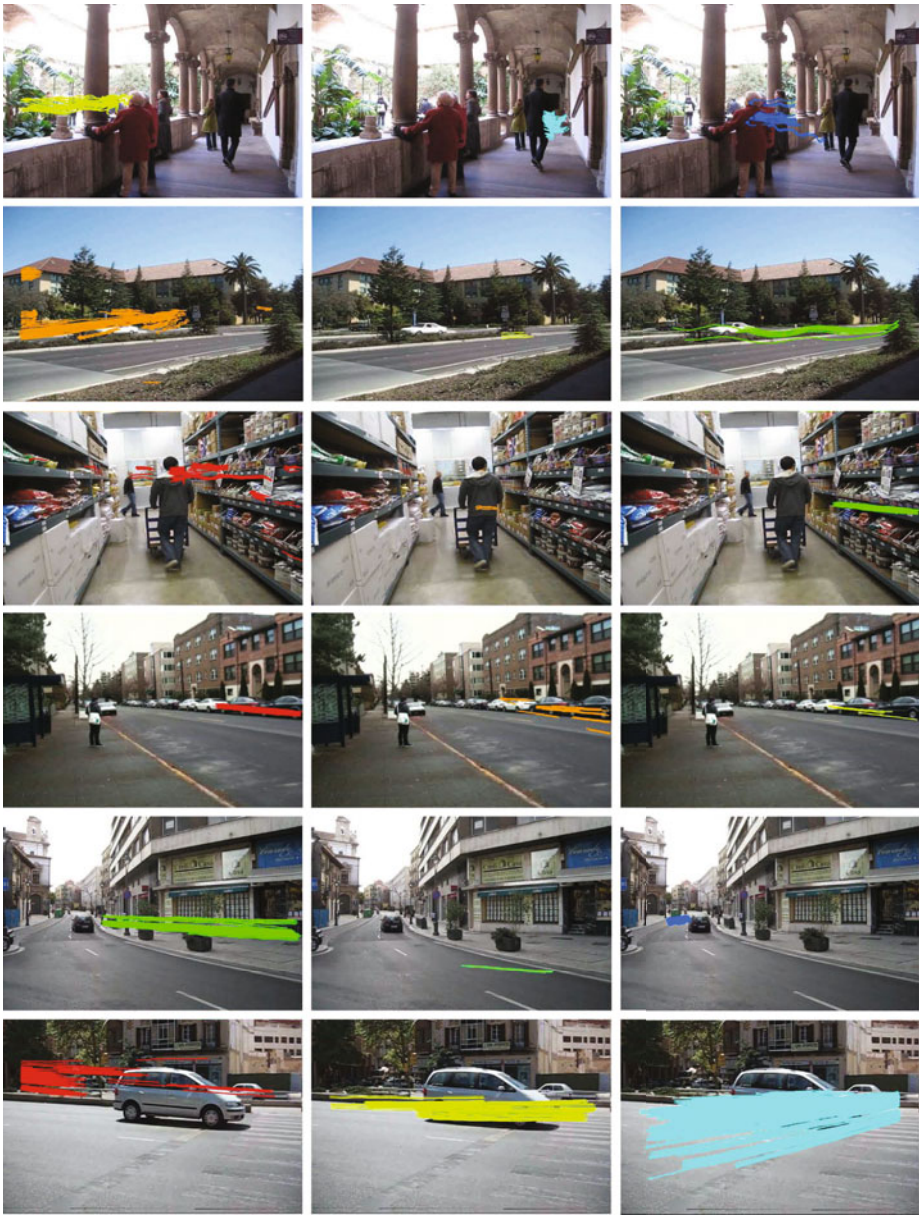


Fig. 5. Event prediction. Each row shows a static image with its corresponding event predictions. For each query image, we retrieve their nearest video clips using scene matching. The events belonging to the nearest neighbors are resized to match the dimensions of the query image and are further clustered to create different event predictions. For example, in a hallway scene, the system predicts motions of different people; in street scenes, it predicts cars moving along the road, etc.

clusters. We try a series of values from 1 to 10 and choose the clustering result that maximizes the distance between clusters. Notice how for different query scenes different predictions that take the image structure are generated.

6.3 Anomaly Detection

Given a video clip, we can also determine if an unusual event is taking place. First, we break down the video clip into query track clusters (which roughly represent object events) using the method described in section 4. We also retrieve the top 200 nearest videos using scene matching. We negatively correlate the degree of anomaly of a query track cluster with the maximum track cluster similarity between the query track cluster and each of the track clusters from the nearest neighbors:

$$anomaly(H_{query}) = -\underset{H_{neigh}}{\text{argmax}}\Big(\mathbf{I}(H_{query}, H_{neigh})\Big) \tag{6}$$

where H_{query} is the spatial histogram of the velocity histories of the query track cluster and H_{neigh} denotes the histogram of a track cluster originated from a nearest neighbor. Intuitively, if we find a similar track cluster in a similar video clip, we consider it as normal. Conversely, a poor similarity score implies that such event (track cluster) does not usually happen in similar video clips. Fig. 6 shows examples of events that our system identified as common by finding a nearest neighbor that minimized its anomaly score. Notice how the nearest track clusters are fairly similar to the query ones and also how the spatial layout of the nearest neighbor scenes matches that of the query video. As a sanity check, notice the similarity of the nearest neighbors average image to the query scene suggesting that the scene retrieval system is picking the right scenes to make accurate predictions. Fig. 7 shows events with a higher anomaly score. Notice how the nearest neighbors differ from the queries. Also, the average images are indicators of noisy and random retrievals. By definition, unusual events will be less likely to appear in our database. However, if the database does not have enough examples of particular scenes, their events will be be flagged as unusual.

Fig. 4(b) shows a quantitative evaluation of this test. Our automatic clustering generates 685 normal and 106 unusual track clusters from our test set. Each of these clusters was scored achieving in similar classification rates when the system is powered by either SIFT or GIST matching methods reaching a 70% detection rate with a 22% false alarm rate. We use the scenario of a random set of nearest neighbors as a baseline. Due to our track cluster distance function, if a cluster similar to the query cluster appears in the random set, our algorithm will be able to identify it and classify the event as common. However, notice that the scene matching methods are demonstrating great utility cleaning up the retrieval set and narrowing videos to a fewer relevant ones. Fig. 8 shows some examples of our system in action.

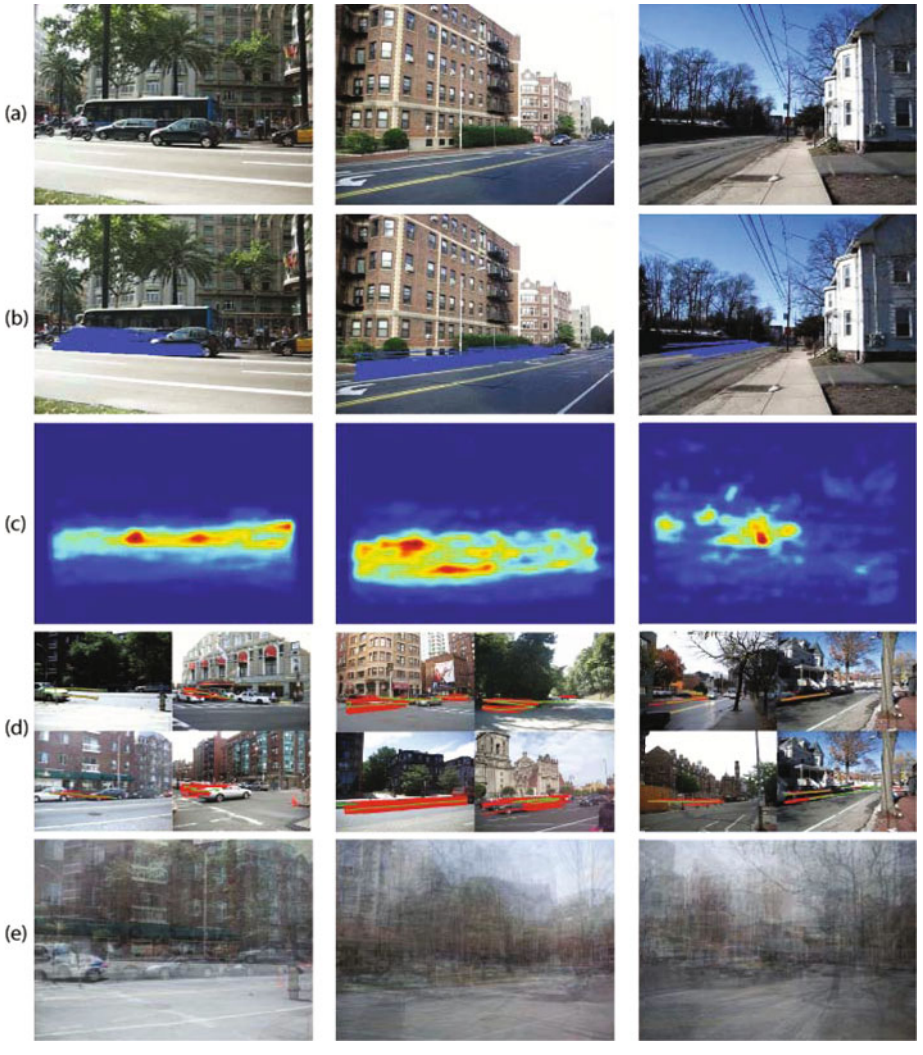


Fig. 6. Track cluster retrieval for common events. A frame from a query video (a), the tracks corresponding to one event in the video (b), the localized motion prediction map (c) generated after integrating the track information of the nearest neighbors (some examples shown in d), and the average image of the retrieved nearest neighbors (e). Notice the definition of high probability motion regions in (c) and how its shape roughly matches the scene geometry in (a). The maps in (c) were generated with no motion information originating from the query videos videos.

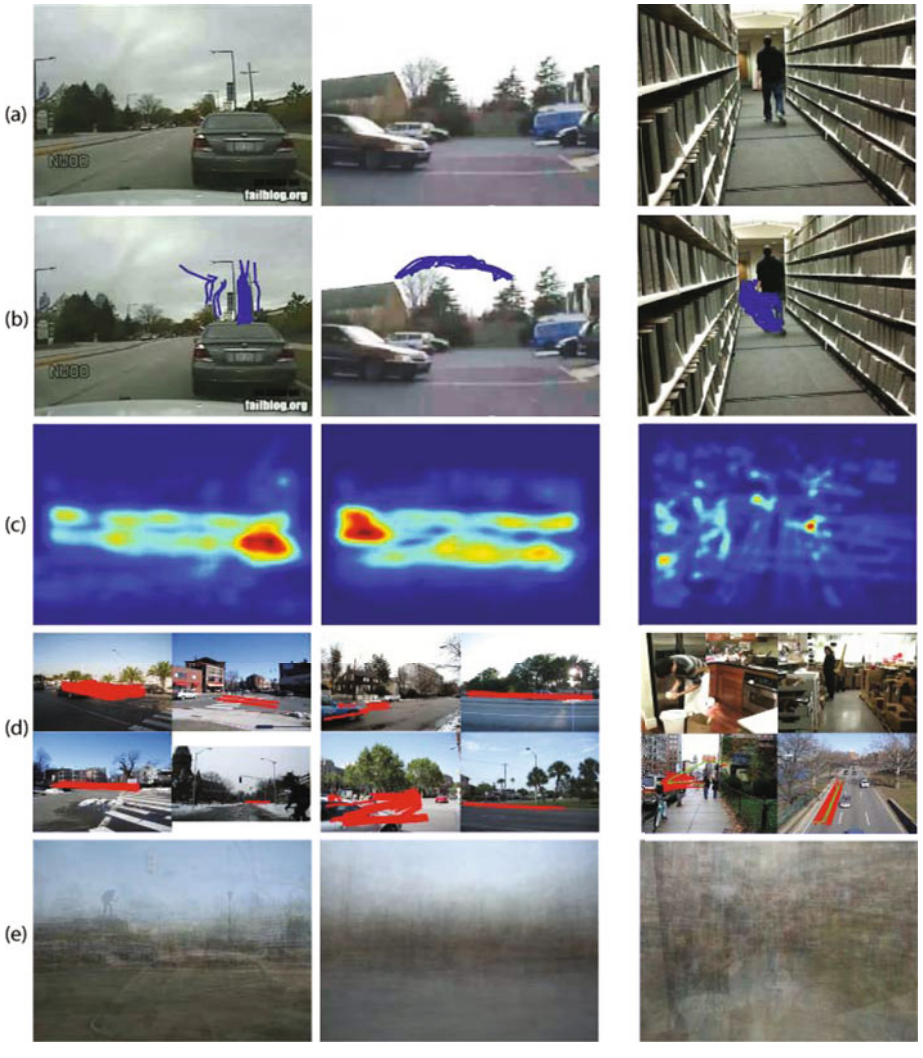


Fig. 7. Track cluster retrieval for unusual events (left) and scenes with less samples in our data set. When presented with unusual events such as a car crashing into the camera or a person jumping over a car while in motion (left and middle columns; key frames can be seen in fig. 8) our system is able to flag these as unusual events (b) due to their disparity with respect to the events taking place in the nearest neighbor videos. Notice the supporting neighbors belong to the same scene class as the query and the motion map predicts movements mostly in the car regions. However, our system fails when an image does not have enough representation in the database (right).

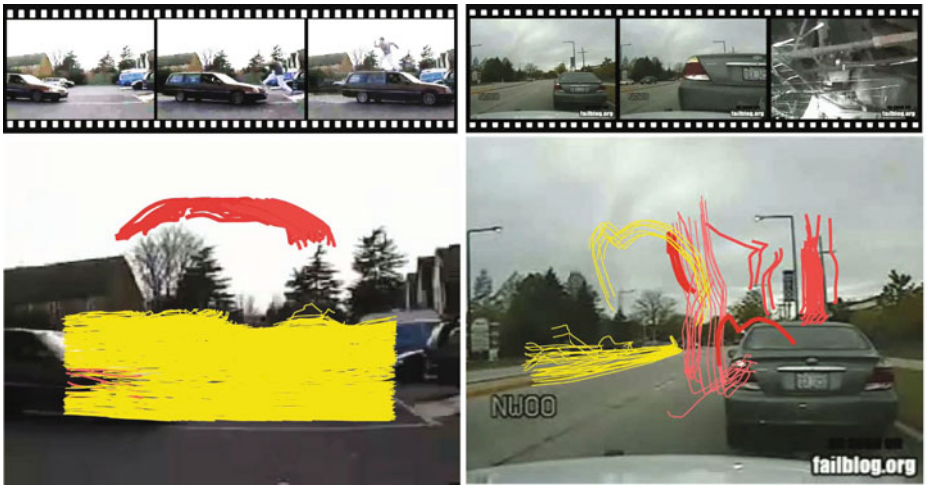


Fig. 8. Unusual event detection. Videos of a person jumping over a car and running across it (left) and a car crashing into the camera (right). Our system outputs anomaly scores for individual events. Common events shown in yellow and unusual ones in red. The thickness and saturation of the red tracks is proportional to the degree of anomaly.

7 Discussion and Concluding Remarks

We have presented a flexible and robust system for unsupervised localized motion prediction and anomaly detection powered by two phases: (1) scene matching to retrieve similar videos given a query video or image, and (2) motion matching via a scene-inspired and spatially aware histogram matching technique for velocity information. We emphasize that most of the work in the literature focuses on action recognition and detection and requires training models for each different action category. Our method has no training phase, is quick, and naturally extends into applications that are not available under other supervised learning scenarios. Experiments demonstrate the validity of our approach when given enough video samples of real world scenes. We envision its applicability in areas such as finding unique content in video sharing websites and future extensions in surveillance applications.

Acknowledgements

This work was funded by NSF Career Award ISI 0747120 and an NDSEG graduate fellowship.

References

1. Winawer, J., Huk, A.C., Boroditsky, L.: A motion aftereffect from still photographs depicting motion. Psychological Science 19, 276–283 (2008)
2. Krekelberg, B., Dannenberg, S., Hoffmann, K.P., Bremmer, F., Ross, J.: Neural correlates of implied motion. Nature 424, 674–677 (2003)

3. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local SVM approach. In: ICPR (2004)
4. Messing, R., Pal, C., Kautz, H.: Activity recognition using the velocity histories of tracked keypoints. In: ICCV. IEEE Computer Society, Washington (2009)
5. Niebles, J.C., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. Int. J. Comput. Vision 79, 299–318 (2008)
6. Laptev, I., Perez, P.: Retrieving actions in movies. In: ICCV (2007)
7. Wang, X., Ma, K.T., Ng, G., Grimson, E.: Trajectory analysis and semantic region modeling using a nonparametric bayesian model. In: CVPR (2008)
8. 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)
9. Junejo, I.N., Javed, O., Shah, M.: Multi feature path modeling for video surveillance. In: International Conference on Pattern Recognition, vol. 2 (2004)
10. Zhong, H., Shi, J., Visontai, M.: Detecting unusual activity in video. In: CVPR (2004)
11. Dalal, N., Triggs, W.: Generalized SIFT based Human Detection. In: CVPR (2005)
12. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for multi-class object layout. In: ICCV (2009)
13. Zanetti, S., Zelnik-Manor, L., Perona, P.: A walk through the web's video clips. In: IEEE Workshop on Internet Vision, associated with CVPR (2008)
14. Efros, A.A., Berg, A.C., Mori, G., Malik, J.: Recognizing action at a distance. In: ICCV (2003)
15. Li, L.J., Fei-Fei, L.: What, where and who? classifying event by scene and object recognition. In: ICCV (2007)
16. Torralba, A., Fergus, R., Freeman, W.: Tiny images. Technical Report AIM-2005-025, MIT AI Lab Memo (September 2005)
17. Liu, C., Yuen, J., Torralba, A.: Nonparametric scene parsing: Label transfer via dense scene alignment. In: CVPR (2009)
18. Hays, J., Efros, A.A.: Scene completion using millions of photographs. In: SIGGRAPH (2007)
19. Hays, J., Efros, A.A.: IM2GPS: estimating geographic information from a single image. In: CVPR (2008)
20. 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)
21. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: ICCV (2003)
22. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. IJCV 42, 145–175 (2001)
23. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR, pp. 2169–2178 (2006)
24. Tomasi, C., Kanade, T.: Detection and tracking of point features. In: IJCV (1991)
25. Birchfield, S.: Derivation of kanade-lucas-tomasi tracking equation. Technical report (1997)
26. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence (2000)
27. Yuen, J., Russell, B.C., Liu, C., Torralba, A.: Labelme video: Building a video database with human annotations. In: ICCV (2009)

Activities as Time Series of Human Postures

William Brendel and Sinisa Todorovic

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

Abstract. This paper presents an exemplar-based approach to detecting and localizing human actions, such as running, cycling, and swinging, in realistic videos with dynamic backgrounds. We show that such activities can be compactly represented as time series of a few snapshots of human-body parts in their most discriminative postures, relative to other activity classes. This enables our approach to efficiently store multiple diverse exemplars per activity class, and quickly retrieve exemplars that best match the query by aligning their short time-series representations. Given a set of example videos of all activity classes, we extract multiscale regions from all their frames, and then learn a sparse dictionary of most discriminative regions. The Viterbi algorithm is then used to track detections of the learned codewords across frames of each video, resulting in their compact time-series representations. Dictionary learning is cast within the large-margin framework, wherein we study the effects of ℓ_1 and ℓ_2 regularization on the sparseness of the resulting dictionaries. Our experiments demonstrate robustness and scalability of our approach on challenging YouTube videos.

1 Introduction

This paper is about efficient, robust, and scalable activity recognition. Our thesis is that certain human actions, such as cycling, diving, walking, and horseback riding, can be compactly represented as short time series of a few still snapshots. Such a discrete activity representation captures discriminative parts of the human body and participating objects (e.g., racquet in playing tennis) in moments when they also assume discriminative postures. Their discriminativeness is defined relative to other human postures and objects seen across different activity classes, so as to allow robust activity recognition. Since there may be only a few time instances in which a few human-body parts strike discriminative poses, the entire space-time volume of a video gets hugely compressed by representing activities as time series. This allows us to develop a robust and scalable, exemplar-based approach to activity recognition in realistic videos with dynamic backgrounds. Numerous video exemplars per activity class can be efficiently stored as time series for the purposes of representing diverse, natural, inter- and intra-class variations. Also, retrieval of exemplars that best match the query can be efficiently done by aligning their short time-series representations.

Our approach consists of the following four computational steps: (1) extracting useful video features, (2) learning a dictionary of discriminative features extracted from a given set of exemplar videos, (3) representing videos as temporal sequences of the

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 721–734, 2010.

learned codewords, and (4) detecting and locating activities in a query video by aligning the query and exemplar time series. In the following, we give an overview of our approach, and point out our main contributions.

Feature Extraction: To represent activities, we extract hybrid features from videos, where the hybrid consists of appearance and local motion cues. Our motivation for using static appearance features comes from the well-known capability of human perception to recognize human actions from still images of activity-characteristic body postures [6, 7]. In cases when different actions (e.g., walking and running) produce similar static features, motion cues that we also extract will help resolve any ambiguity about static features. Prior work also often combines local motion and static features [1, 2, 3, 4, 5], since their extraction is reportedly more robust than that of other types of features, such as 2D+t volumes, optical flow, etc. We segment each video frame by the standard hierarchical meanshift algorithm, as in [8]. Meanshift regions are described by the HOG descriptor [9], shown to be stable and discriminative under a certain amount of partial occlusion, and changes in object pose [10]. HOG is computed using the spatial derivative of pixel intensities in the frame. HOG's are invariant to similar camera motions (e.g., panning) across videos, which may produce similar motion features of distinct actions. We also compute the temporal derivative of pixel intensities between two frames, resulting in the 2D+t HOG descriptor associated with every meanshift region.

Dictionary Learning: Given a large set of 2D+t HOG's, extracted from all exemplar videos, we learn a sparse dictionary of codewords, each representing the most discriminative 2D+t HOG's in the set. Since the HOG's are anchored at meanshift regions of video frames, the learned codewords may correspond to the entire human body, or body part, as well as to an object taking part in the activity (e.g., horses head in horseback riding, or swing in swinging). Existing work typically clusters video features by K-means [1, 11, 12], yielding codewords that may not be relevant for discriminating among the action classes. There is very little work on dictionary leaning for activity recognition, with few exceptions [13, 4]. Their information-bottleneck formulation, however, is intractable and requires approximation, which may not learn the optimal dictionary. In contrast, we cast dictionary learning within the large-margin framework, and derive an efficient, linear-complexity algorithm, with strong theoretical guarantees of small generalization error. Another key difference from prior work is that our codewords may represent objects defining the activity, in addition to human-body parts. This is critical for differentiating between very similar activities in which the human body undergoes similar motions but interacts with different objects (e.g., eating a banana vs. answering the phone). Most existing methods, however, do not account for objects that people interact with while performing the activity. This is because they seek to crop out only people from the videos by various means of background subtraction [14], or by applying people detectors [2, 12]. Recent studies show that activity recognition may improve when co-occurring objects in the context are identified [11]. Unlike all previous work, we use only video labels, i.e., weak supervision, for our dictionary learning.

Time-Series Representation: We represent videos as temporal sequences of codewords of the dictionary, learned in the previous step. Given a video, its time series

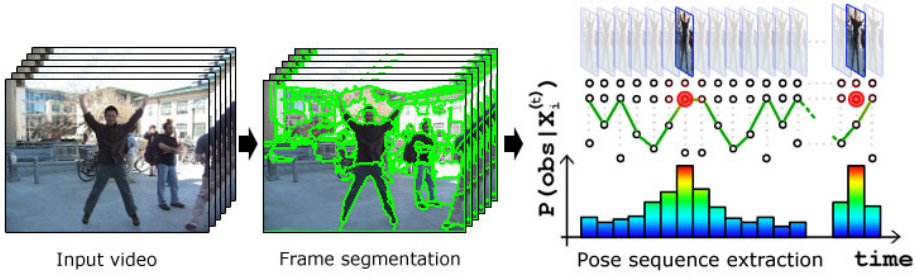


Fig. 1. Compact video representation: Meanshift regions, extracted from video frames, are matched with the codewords representing discriminative human-body parts and activity-defining objects. Best matching candidates are tracked across the frames by the Viterbi algorithm, resulting in a short time series of a few discriminative, still snapshots (marked red).

is computed by tracking candidate detections of the codewords in each frame, as illustrated in Fig. 1. For this tracking, we use the Viterbi algorithm which sequentially pursues the bast track at any given state, defined by a product of all codewords and meanshift segments in the visited frame. The codewords carry the information about their relative time locations in the exemplar videos from which they have been extracted. This allows the Viterbi algorithm to enforce the activity-characteristic temporal consistency of the resulting time-series representation. Prior work also seeks to represent videos as sequences of shape-motion prototypes [12]. However, they detect the prototypes in each frame, and thus generate long sequences of prototypes spanning all frames. Also, their prototypes represent the entire human body, giving our part-based codewords advantage in the presence of partial occlusions.

Recognition: Given a query video, and its time-series representation, it is aligned with the exemplar sequences by the cyclic dynamic time warping (CDTW) [15]. The activity label of the best aligned exemplar is transferred to the query, where their CDTW alignment also localizes the activity's occurrence in the space-time volume of the query. As shown in Sec. 5, we achieve the average recognition rate of 77.8% on challenging YouTube videos, outperforming the state-of-the-art result of 71.2% from [4].

Our Contributions include: (i) Four alternative, weakly supervised methods for learning a sparse dictionary of video features, formulated within the large-margin framework, using ℓ_1 or ℓ_2 regularizations; (ii) Proofs that the four methods converge to their respective globally optimal solutions, subject to the four distinct objective functions considered; (iii) Accounting for the co-occurrence statistics of objects and human actions in the scene, and thus extracting discriminative objects, which participate in the activity, along with discriminative human postures; and (iii) Robust and scalable exemplar-based approach to activity detection and localization in videos.

In the following, Sec. 2 explains the video features we use, Sec. 3 presents the four algorithms for dictionary learning and proofs of their convergence, Sec. 4 describes how to extract and align the time-series representations of videos for activity recognition, and Sec. 5 presents our experimental evaluation.

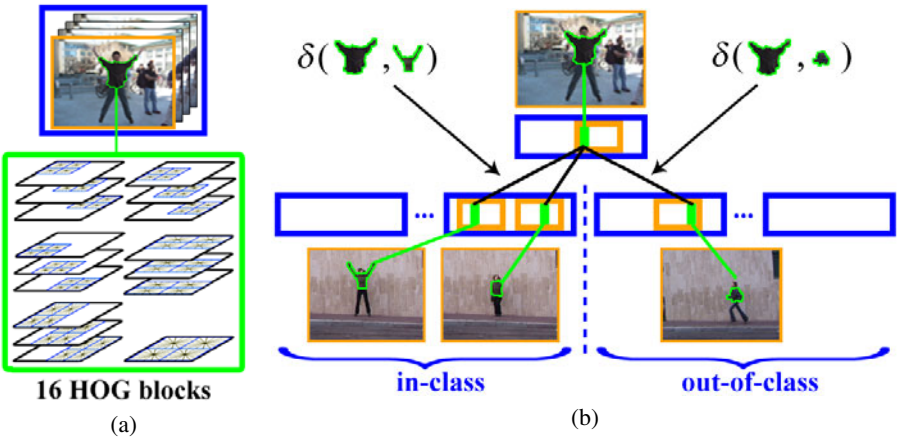


Fig. 2. (a) The meanshift regions (green) of all frames (orange) in a video (blue) are characterized by the 2D+t HOG descriptors, called hybrid features as they combine static appearance and motion cues. The 2D+t HOG of a meanshift region uses orientations of spatial and temporal gradients of pixel intensities, extracted from 16 overlapping windows covering the region. (b) Computing distances between in-class and out-of-class videos. (best viewed in color).

2 Feature Extraction

This section specifies appearance and local motion features that we use in this paper. Each frame is first partitioned into segments using the standard hierarchical meanshift algorithm, as in [8]. The segments provide static appearance features, at multiple scales. Each meanshift region is then described using a 2D+t HOG descriptor, which additionally incorporates local motion cues. The 2D+t HOG extends the standard HOG [9], which has been shown to exhibit invariance to partial occlusion and object deformations [10]. We first use the difference operators in time and space to compute the 2D+t gradient vectors at every pixel of the meanshift region. Then, we project these 3D vectors onto the *x-y*, *x-t*, and *y-t* planes. Next, each projection is covered by 16 overlapping blocks, as shown Fig. 2a. From each block we extract a 36-dimensional histogram of oriented gradients (9 bins for 4 cells within one block). By concatenating the three 36D histograms from *x-y*, *x-t*, and *y-t* planes, we obtain the 2D+t HOG with 108 dimensions.

3 Learning the Dictionary of Activity Codewords

In this section, we specify four alternative algorithms for learning the dictionary of discriminative activity features, and present their convergence analysis. We begin by introducing some notation and basic definitions. Suppose that we are given a set of annotated exemplar videos $\mathbb{D} = \{\boldsymbol{x}_i, y_i\}$, where $\boldsymbol{x}_i = [\boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{ik}, \ldots]^{\mathrm{T}}$ denotes all 2D+t HOG's extracted from all frames of video i, and y_i is the associated label of activity class. Note that different videos may have different total numbers of features.

Our goal is to identify the most discriminative features in the entire set $\mathbb{D}$, called codewords. In this paper, we consider learning two types of dictionaries. If the codewords are learned so a given class is discriminated well against the other classes, we obtain the dictionary of that class. If the codewords are learned to discriminate well all classes, they form the all-class dictionary.

We formulate dictionary learning within the large-margin framework. Margins play a crucial role in the modern machine learning [17]. They measure the confidence of a classifier when making a decision. There are two types of margins. The more common type, called sample-margin, used for example in SVMs, measures how far positive and negative training examples are separated by the decision surface. In this paper, we consider the other type called hypothesis-margin. It is defined per data instance, and measures a distance between the hypothesis and the closest hypothesis that assigns alternative label to that instance. In particular, for each $\boldsymbol{x}_i$, we seek to maximize its distance to all out-of-class videos, called misses. At the same time, we wish to minimize its distance to all videos belonging to the same class, called hits. These two objectives can be achieved by maximizing the hypothesis-margin of the one-nearest-neighbor classifier (1NN). Maximizing the sample-margin of the SVM has been used for dictionary learning in [16]. However, this formulation, is not suitable for videos, since it would lead to a large scale optimization problem of prohibitive complexity.

To specify the hypothesis-margin of 1NN, we define an asymmetric distance between two videos, $d_{ij} = d(\boldsymbol{x}_i, \boldsymbol{x}_j)$, as a weighted sum of distances between their best matching features, $d_{ij} = \boldsymbol{\delta}_{ij}^{\mathrm{T}} \boldsymbol{w}_i$. The vector $\boldsymbol{\delta}_{ij} = [\delta_{ij1}, \ldots, \delta_{ijk}, \ldots]^{\mathrm{T}}$ consists of χ^2 distances between the histograms of each 2D+t HOG descriptor, x_{ik}, and its best matching descriptor in $\boldsymbol{x}_j$, $\delta_{ijk} = \min_l \chi^2(\boldsymbol{x}_{ik}, \boldsymbol{x}_{jl})$. The non-negative weights, $\boldsymbol{w}_i \geq \mathbf{0}$, and distances $\boldsymbol{\delta}_{ij}$ are associated with features of the first video in the pair, $\boldsymbol{x}_i$, and thus $\boldsymbol{x}_i$, $\boldsymbol{w}_i$, and $\boldsymbol{\delta}_{ij}$ have the same length. Note that the weights $\boldsymbol{w}_i$ serve to indicate the relevance of the corresponding features in $\boldsymbol{x}_i$ for discriminating between activity classes y_i and y_j. Our goal is to learn $\boldsymbol{w}_i$ for all videos $\boldsymbol{x}_i$, so as to maximize the hypothesis-margin of 1NN, and then extract video features with the highest weights to the dictionary. We specify the hypothesis-margin of specific $\boldsymbol{x}_i$ as

$$\rho_i = d_{im} - d_{ih} = (\boldsymbol{\delta}_{im} - \boldsymbol{\delta}_{ih})^{\mathrm{T}} \boldsymbol{w}_i \ , \tag{1}$$

where index m denotes that $\boldsymbol{\delta}_{im}$ is computed with the nearest miss of $\boldsymbol{x}_i$, and index h denotes that $\boldsymbol{\delta}_{ih}$ is computed with the nearest hit of $\boldsymbol{x}_i$. From (1), it follows that maximizing the hypothesis-margin of 1NN will amount to maximizing the distances of all videos from their respective out-of-class videos, and simultaneously minimizing the distances of all videos to their respective in-class videos. This can be formulated using the following notation. Let $\boldsymbol{w}$ be a column vector of concatenated weights $\boldsymbol{w}_i$ for all $\boldsymbol{x}_i \in \mathbb{D}$; $\boldsymbol{z}_m$ be a column vector of concatenated feature distances $\boldsymbol{\delta}_{im}$ for all $\boldsymbol{x}_i \in \mathbb{D}$ to their respective nearest misses; and $\boldsymbol{z}_h$ be a column vector of concatenated feature distances $\boldsymbol{\delta}_{ih}$ for all $\boldsymbol{x}_i \in \mathbb{D}$ to their respective nearest hits. Finally, let $\boldsymbol{z} = \max(\mathbf{0}, \boldsymbol{z}_m - \boldsymbol{z}_h)$. Then, dictionary learning can be specified as the following linear program (LP):

$$\underset{\boldsymbol{w}}{\operatorname{argmax}} \ \boldsymbol{z}^{\mathrm{T}} \boldsymbol{w}, \quad \text{s. t. } \boldsymbol{w} \geq \mathbf{0}, \ \text{ and } \ \|\boldsymbol{w}\| \leq \gamma \ , \tag{2}$$

where γ is a positive constant, and $\|\cdot\|$ is either ℓ_1 or ℓ_2 norm. After solving (2), features with non-zero weights will be selected as codewords in the dictionary.

When $\boldsymbol{w}$ and $\boldsymbol{z}$ represent the concatenation of feature weights and distances across all videos, the resulting dictionary will be all-class. Similarly, the dictionary of a specific class can be derived by concatenating into $\boldsymbol{w}$ and $\boldsymbol{z}$ the appropriate values for only those videos that belong to that class.

Note that (2) represents an extremely large optimization problem. Any naive use of general LP solvers, such as simplex or interior point methods, would be computationally too expensive. In the sequel, we propose four alternative algorithms to solve (2), which are very efficient, with linear complexity in the number of input video features.

3.1 Logistic-Regression Formulation

In this subsection, we employ the logistic regression formulation to solve our large LP problem, given by (2). Specifically, to eliminate the constraint $\|\boldsymbol{w}\| \leq \gamma$ from (2), we add a penalty term, $\lambda \|\boldsymbol{w}\|$, directly to the objective function, where λ is a non-negative input parameter. Note, however, that the objective function of (2) represents maximization, whereas the constraint $\|\boldsymbol{w}\| \leq \gamma$ requires minimization. This can be resolved by reformulating (2) as

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} \ \log[1+\exp(-\boldsymbol{z}^{\mathrm{T}}\boldsymbol{w})] + \lambda \|\boldsymbol{w}\|, \ \text{ s. t. } \ \boldsymbol{w} \geq 0 \ . \tag{3}$$

Note that λ controls the sparseness of the solution, and thus the number of selected codewords in the dictionary.

Eq. (3) is a constrained convex optimization problem. Due to the non-negative constraint on $\boldsymbol{w}$, it cannot be solved directly by gradient descent. To overcome this difficulty, we use the following substitution $\boldsymbol{w} = [v_1^2, \ldots, v_k^2, \ldots]^{\mathrm{T}}$, where v_k are auxiliary variables, and k is the index over all 2D+t HOG's. This gives

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} \ \log[1+\exp(-\textstyle\sum_k z_k v_k^2] + \lambda R(\boldsymbol{v}), \tag{4}$$

where $R(\boldsymbol{v}) = \|\boldsymbol{v}\|_2^2$ for ℓ_1 regularization, or $R(\boldsymbol{v}) = \sqrt{\sum_k v_k^4}$ for ℓ_2 regularization. Consequently, we obtain an unconstrained optimization problem. It is straightforward to derive the following gradient-descent solution of (4):

$$LR\text{-}\ell_1: \ \ v_k \leftarrow v_k - \eta \left(\lambda - \frac{\exp(-\sum_k z_k v_k^2)}{1+\exp(-\sum_k z_k v_k^2)} \right) \cdot v_k, \qquad \text{for } \ell_1, \tag{5}$$

$$LR\text{-}\ell_2: \ \ v_k \leftarrow v_k - \eta \left(\lambda \frac{v_k^2}{\sqrt{\sum_k v_k^4}} - \frac{\exp(-\sum_k z_k v_k^2)}{1+\exp(-\sum_k z_k v_k^2)} \right) \cdot v_k, \quad \text{for } \ell_2, \tag{6}$$

where η is the learning rate determined by the standard line search. Once v_k are estimated, we then compute the feature relevances as $w_k = v_k^2$, $k = 1, 2, \ldots$. The convergence of this logistic-regression based algorithm is explained at the end of this section, after we specify the other two alternative algorithms.

3.2 Alternative LP Formulation

In practice, the update rules given by (5) and (6) have a serious limitation. In particular, if the term $\sum_k z_k v_k^2$ is large, then $\exp(-\sum_k z_k v_k^2)$ drops exponentially to zero, and the update depends only on the penalty term. To overcome this problem, we modify the LP given by (2), as follows.

Without a loss of generality, we replace the constraint $\|\boldsymbol{w}\| \leq \gamma$ by $\|\boldsymbol{w}\| = \gamma$, leading to the following new LP formulation

$$\underset{\boldsymbol{w}}{\operatorname{argmax}}\ \boldsymbol{z}^{\mathrm{T}} \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|},\ \text{s. t. }\ \boldsymbol{w} \geq 0. \tag{7}$$

As in Sec. 3.1, the non-negative constraint in (7) can be reformulated by using the following substitution $\boldsymbol{w} = [v_1^2, \ldots, v_k^2, \ldots]^{\mathrm{T}}$, where v_k are auxiliary variables, and k is the index over all video features. This gives

$$\underset{\boldsymbol{w}}{\operatorname{argmax}}\ \frac{1}{R(\boldsymbol{v})} \sum_k z_k v_k^2, \qquad \boldsymbol{w} = [v_1^2, \ldots, v_k^2, \ldots]^{\mathrm{T}}, \tag{8}$$

where, as in (4), $R(\boldsymbol{v}) = \|\boldsymbol{v}\|_2^2$ for ℓ_1, or $R(\boldsymbol{v}) = \sqrt{\sum_k v_k^4}$ for ℓ_2 regularization. It is straightforward to derive the following gradient-ascent solution of (8):

$$LP\text{-}\ell_1: \quad v_k \leftarrow v_k + \eta \frac{\left(z_k \sqrt{R(\boldsymbol{v})} - \sum_k z_k v_k^2\right)}{R(\boldsymbol{v})} \cdot v_k, \qquad \text{for } \ell_1, \tag{9}$$

$$LP\text{-}\ell_2: \quad v_k \leftarrow v_k + \eta \frac{\left(z_k \sqrt{R(\boldsymbol{v})} - \frac{v_k^2}{R(\boldsymbol{v})} \sum_k z_k v_k^2\right)}{R(\boldsymbol{v})} \cdot v_k, \qquad \text{for } \ell_2, \tag{10}$$

where η is the learning rate determined by the standard line search. Once v_k are estimated, we then compute the feature relevances as $w_k = v_k^2$, $k = 1, 2, \ldots$.

Convergence: In both LP formulations, presented in Sec. 3.1 and 3.2, we reformulate the non-negative constraints in (3) and (7). The resulting objective functions, given by (4) and (8), are convex and concave, respectively. Consequently, the gradient descent in (5)–(6), and the gradient ascent in (9)–(10) converge to their respective globally optimal solutions. The full proof that (4) is convex, and (8) is concave is given in the supplemental material. The proof first shows that the substitution $w_k = v_k^2$, $k = 1, 2, \ldots$, does not change the concavity of the original LP formulation, given by (2). Then, we use the classical theoretical results in convex optimization about the convexity and concavity of a composition of two functions $(f \circ g)$ to prove that the logistic regression formulation is convex, and the alternative normed objective is concave.

Complexity of both formulations presented in Sec. 3.1 and 3.2 is linear in the number of input video features. Since our features are descriptors of meanshift segments, the total number of our features is significantly smaller than interest-point features, typically used in existing approaches to activity recognition.

After convergence, all 2D+t HOG's from all videos in $\mathbb{D}$ whose weights are nonzero are declared as codewords. Finally, the 2D+t HOG descriptor of each codeword is augmented with the time stamp of a frame from which the codeword has been extracted, normalized relative to the length of the originating video. This is used to enforce the temporal consistency of codewords along time series representing videos.

4 Representing Videos as Time Series of Activity Codewords

This section describes how to compute the time-series representation of a video. We first extract multiscale meanshift regions in each video frame, and then match their 2D+t HOGs with the codewords. The standard Viterbi algorithm is applied to track the best matches (Fig. 1), where for each frame only one best matching codeword-region pair is selected. Tracking seeks to maximize the joint likelihood of all matches along the Viterbi path, under the constraint that the tracked codewords along the path are locally smooth in the 2D space, and temporally consistent. In the following, we specify the Viterbi algorithm.

Let $\Omega = \{\boldsymbol{\omega}_l\}_{l=1,2,...}$ denote the dictionary of activity codewords, and let $\boldsymbol{x}^{(t)} = \{\boldsymbol{x}_k^{(t)}\}_{k=1,2,...}$ denote 2D+t HOG's extracted from frame t of video $\boldsymbol{x}$. In each frame t, the goal of the Viterbi algorithm is to select a single, best matching pair $(\boldsymbol{x}_k^{(t)}, \boldsymbol{\omega}_l)$ out of the entire product space $\boldsymbol{x}^{(t)} \times \Omega$. The selected, unique pair $(\boldsymbol{x}_k^{(t)}, \boldsymbol{\omega}_l)$ is referred to as instantiation of codeword $\boldsymbol{\omega}_l$ in frame t, and denoted as $\hat{\boldsymbol{\omega}}_l^{(t)} = (\boldsymbol{x}_k^{(t)}, \boldsymbol{\omega}_l)$. Across all frames, the goal of the Viterbi algorithm is to satisfy the temporal constraints between the instantiated codewords $\{\hat{\boldsymbol{\omega}}_l^{(t)}\}_{t=1,2,...}$, and produce a locally smooth trajectory in the 2D space. Temporal consistency is enforced via a Markov chain which is informed by the time stamps associated with codewords, as mentioned in the previous section. To formalize the above two goals of the Viterbi algorithm, we below first specify the likelihood that measures the quality of matches between video features and codewords, and then define the transition probability of the Markov chain which favors spatially smooth and temporally consistent codeword instantiations from one frame to another.

Video feature $\boldsymbol{x}_k^{(t)}$ matches codeword $\boldsymbol{\omega}_l$ with likelihood $P(\boldsymbol{x}_k^{(t)}|\boldsymbol{\omega}_l) \propto e^{-\alpha\chi^2(\boldsymbol{x}_k^{(t)}, \boldsymbol{\omega}_l)}$, where $\alpha = 0.01$ (empirically found at equal error rate) weights the χ^2 histogram distance (for equal error rate, we get). The Markov-chain transition probability is defined as $P(\hat{\boldsymbol{\omega}}_l^{(t)}|\hat{\boldsymbol{\omega}}_j^{(t-1)}) \propto e^{-\boldsymbol{\beta}^{\mathrm{T}}|\hat{\boldsymbol{\omega}}_l^{(t)} - \hat{\boldsymbol{\omega}}_j^{(t-1)}|}$, where $\boldsymbol{\beta} = [0.1, 0.1]^{\mathrm{T}}$ (empirically found at equal error rate), and $|\cdot|$ denotes the absolute difference of the corresponding spatial and time coordinates of the instantiated codewords $\hat{\boldsymbol{\omega}}_l^{(t)}$ and $\hat{\boldsymbol{\omega}}_j^{(t-1)}$. Specifically, for their spatial coordinates, we take the centroids of meanshift regions that got matched to $\boldsymbol{\omega}_l$ and $\boldsymbol{\omega}_j$ in frames t and $(t-1)$. For their time coordinates, we take the time stamps that $\boldsymbol{\omega}_l$ and $\boldsymbol{\omega}_j$ carry from their source exemplar videos. With these definitions, we specify the Viterbi algorithm as finding the optimal sequence of codewords so the following Markov chain is maximized:

$$P(\hat{\boldsymbol{\omega}}_l^{(t)}) = \max_{\hat{\boldsymbol{\omega}}_j^{(t-1)},\ \boldsymbol{x}_k^{(t)}} P(\hat{\boldsymbol{\omega}}_j^{(t-1)}) P(\hat{\boldsymbol{\omega}}_l^{(t)}|\hat{\boldsymbol{\omega}}_j^{(t-1)}) P(\boldsymbol{x}_k^{(t)}|\hat{\boldsymbol{\omega}}_l^{(t)}), \tag{11}$$

where $P(\hat{\boldsymbol{\omega}}_j^{(t-1)})$ is recursively defined. The Viterbi algorithm retrieves the best path across the frames (Fig. 1) with linear complexity in the number of video features.

Extracting the Compact Representation: The obtained Viterbi path is characterized by a sequence of likelihoods $P(\boldsymbol{x}_k^{(t)}|\hat{\boldsymbol{\omega}}_l^{(t)})$, $t = 1, 2, \ldots$. This sequence has modes and valleys, as illustrated in Fig. 1. The valleys indicate low confidence in the corresponding codeword instantiations. We identify and eliminate the valleys in this likelihood

sequence by the popular quick-shift mode-seeking algorithm [18]. As a result, we obtain the compact time-series representation.

Exemplar-based Recognition: Given a query video, we use the same algorithm to extract its time series of codewords. For recognition, we align the time series of the query and exemplar videos. Note that the sought activity may not start at the beginning, or finish at the end of the query video. Therefore, the query-exemplar alignment is not only aimed at finding the best matching exemplar, but also to localize a subsequence of codewords, in the query time series, that represents the activity. The label of the best aligned exemplar is taken as the activity class of the query. Also, the codewords identified to represent the activity in the time series are back-tracked to the space-time locations of the corresponding meanshift regions in the query video. All this results in the simultaneous detection and localization of the activity in the query video. In this paper, two temporal sequences of codewords are aligned by the cyclic dynamic time warping (CDTW), presented in [15]. CDTW finds correspondences between codewords of the two sequences by identifying the optimal path in a cost matrix of all pairwise codeword matches. This is done by respecting the ordering of each input sequence. The costs are χ^2 distance between the 2D+t HOG histograms of each codeword. We use the cyclic variant of DTW, because it efficiently identifies the optimal start and end of the alignment path in the cost matrix, regardless of the lengths of the input sequences. Complexity of CDTW is linear in the total number of elements in the two sequences.

5 Results

Experiments are conducted on five benchmark datasets: Weizmann activities [14], KTH [19], UM "Gestures" [12], CMU "Crowded" videos [8], and UCF "YouTube" [4]. KTH contains a varied set of challenges, including scale changes, variation in the speed of activity execution, and indoor and outdoor illumination variations. In UM "Gestures", training videos are captured by a static, high-resolution camera, with the person standing in front of a uniform background; whereas test videos are captured by a moving camera, in the presence of a background clutter, and other moving objects. The CMU "Crowded" videos are acquired by a hand-held camera, in unconstrained environments, with moving people or cars in the background. Each CMU video may contain several target actions, where we identify only one. This dataset is challenging due to significant spatial- and temporal-scale differences in how the subjects perform the actions. In the UCF "YouTube" videos the actors interact with objects, such as a horse, bicycle, or dog, which define the corresponding activities. This dataset is challenging due to: a mix of steady and shaky cameras, cluttered background, low resolution, and variations in scale, viewpoint, and illumination.

For activity recognition, we use 5 exemplars per each class from the considered dataset. The activity class of a given query is defined by a majority voting of M, best-aligned exemplars, where M is estimated by the leave-one-out (LOO) strategy. We report the average classification accuracy at equal error rate (EER), where the accuracy is

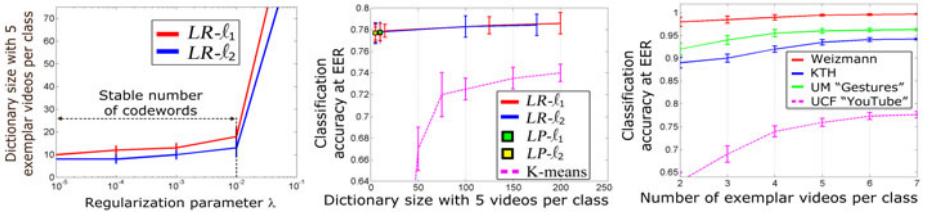


Fig. 3. (a) Average dictionary size per activity class in the UCF "YouTube" dataset as a function of the regularization parameter λ in LR-ℓ_1 and LR-ℓ_2. (b) Classification accuracy at EER averaged over the UCF "YouTube" classes vs. the average size of the dictionary generated by LR-ℓ_1, LR-ℓ_2, LP-ℓ_1, LP-ℓ_2, and unsupervised K-means clustering of 2D+t HOGs. (c) Average classification accuracy on all datasets vs. the number of available exemplar videos, when the dictionary is learned by LP-ℓ_1. (best viewed in color).

averaged over all classes in the dataset. On all datasets, we achieve EER for input parameters $\lambda = 10^{-3}$, $\alpha = 0.01$, and $\boldsymbol{\beta} = [0.1, 0.1]$. In the following, we present evaluation of the individual steps of our approach.

Dictionary Learning: In the following two experiments, we use the UCF "YouTube" dataset to extract distinct dictionaries for each class (not the all-class dictionary). First, we evaluate our sensitivity to the specific choice of λ in LR-ℓ_1 and LR-ℓ_2. Fig. 3a shows the average dictionary size as a function of input λ values, where each dictionary is learned on five exemplar videos per class, and the dictionary size is averaged over all "YouTube" classes. As can be seen, for a wide range of λ values, when $\lambda < 10^{-2}$, both LR-ℓ_1 and LR-ℓ_2 produce a "stable" number of codewords. Second, we evaluate our classification accuracy at equal error rate (EER) versus the average size of different dictionary types produced by LR-ℓ_1, LR-ℓ_2, LP-ℓ_1, LP-ℓ_2, as well as by unsupervised K-means clustering. Fig. 3b shows that all our learning methods outperform the unsupervised clustering of video features by K-means. As can be seen in Fig. 3b, when using all four learning methods we achieve similar classification accuracy.

Depending on a particular application, one may prefer to work with the dictionary generated by LP-ℓ_1, because LP-ℓ_1 yields the sparsest solution with the fewest codewords, and it does not require any input parameter (unlike LR-ℓ_1 and LR-ℓ_2). Therefore, in the following, we continue with evaluation of our approach when using only LP-ℓ_1 for dictionary learning.

Accuracy vs. Number of Exemplars: We test our performance on each dataset versus the number of randomly selected exemplar videos per class. Classification accuracy is averaged over all classes within the specific dataset. Fig. 3c shows that only few exemplars are needed to achieve high accuracy for the challenging datasets.

HOG vs. 2-D+t HOG: We test whether adding motion cues to the standard HOG increases performance. Fig. 4 shows that our performance on the UCF "YouTube" videos is better with 2D+t HOG's than that with HOG's, since the additional motion features help disambiguate similar static appearance features.

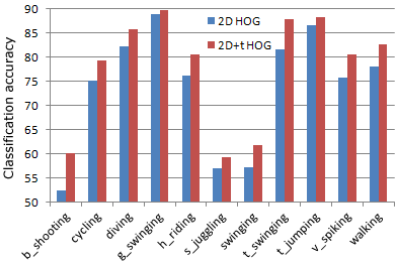


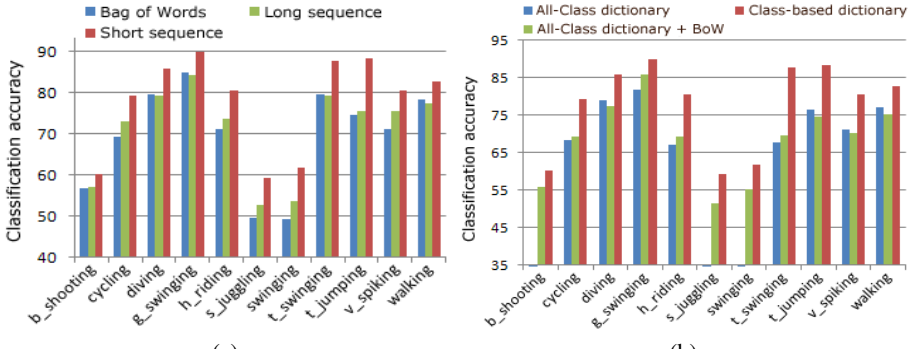
Fig. 4. Classification accuracy at EER when using HOG's and 2D+t HOG's on the UCF "YouTube" dataset, for LP-ℓ_1

Table 1. Recall of detecting relevant video parts for activity recognition by our Viterbi algorithm, evaluated with respect to the manually annotated bounding boxes around actors, and averaged over all videos and classes

	Recall
Weizmann	0.95
KTH	0.94
UM "Gestures"	0.95

Viterbi-based Codeword Tracking: We evaluate recall of our Viterbi-based detection of relevant video parts for activity recognition. To this end, we use the ground-truth bounding boxes around actors, provided in the Weizmann, KTH and UM "Gestures" datasets. Ideally, the Viterbi algorithm would associate codewords with those meanshift regions that fall within the bounding boxes in every video frame. We estimate recall as a ratio between the number of true positives and the total number of frames, where a true positive is a detected meanshift region with more than 50% of its area falling within the bounding box. Our recall averaged over all videos and classes is shown in Table 1.

Viterbi vs. Bag-of-Words: Tracking codewords by the Viterbi algorithm increases complexity vs. a simpler Bag of Words (BoW) approach, which scans all meanshift regions, and finds the best matching region-codeword pair, in each frame, irrespective of the results in other frames. The increased complexity is justified by significant increase in our classification accuracy vs. BoW, as shown in Fig. 5a.



(a) (b)

Fig. 5. Classification accuracy at EER of Bag-of-words, and our approach with LP-ℓ_1, on the UCF "YouTube" dataset: (a) Our approach uses short time series, and long sequence of codewords as the video representation. The short time series enables faster and more accurate activity recognition. (b) Our approach uses the all-class dictionary and a set of dictionaries learned per class. The class-based dictionary learning gives better performance.

Table 2. AUC for CMU "Crowded" videos

	[3]	[8]	Ours (LP-ℓ_1)
pick-up	0.58	0.47	0.60
one-hand wave	0.59	0.38	0.64
jumping jack	0.43	0.22	0.45
two-hands wave	0.43	0.64	0.65

Table 3. Average classification accuracy at EER

	[14]	[12]	[4]	[3]	Ours (LP-ℓ_1)
Weizmann	97.5	X	X	X	99.7
KTH	X	95.7	91.8	87.8	94.2
UM "Gestures"	X	95.2	X	X	96.3
UCF "YouTube"	X	X	71.2	X	77.8

Long vs. Short Time Series: After the Viterbi algorithm has identified the optimal path of codewords in a video, we eliminate a number of codeword detections with low confidence, and thus extract the short time series representation. Fig. 5a shows significant performance gains, on the the "YouTube" dataset, when using the short time series vs. the long sequence of codewords instantiated in every video frame, as the video representation. In addition, the short time series enable nearly two-orders-of-magnitude speed ups of recognition. On the "YouTube" videos, recognition by aligning long sequences (whose size is the same as the number of video frames) takes on average 302.2ms, whereas short time series are aligned in only 4.6ms. Our implementation is in C on 2.8GHz 8GB RAM PC.

All-class dictionary vs. class-based dictionaries: Fig. 5b compares our performance, when using a set of dictionaries learned per class vs. the all-class dictionary. As can be seen, the all-class dictionary yields inferior performance. This is because the all-class dictionary is typically very sparse, so that an activity class may not be even represented by any codeword (see 'b_shooting', 's_juggling' and 'swinging'). Interestingly, for a few classes, BoW with the all-class dictionary outperforms our approach with the all-class dictionary.

Training Transfer: We evaluate whether our approach can be trained on a simple, sanitized setting of the Weizmann videos, and then used for activity recognition on the challenging CMU "Crowded" videos. Specifically, we use 5 exemplar videos per class

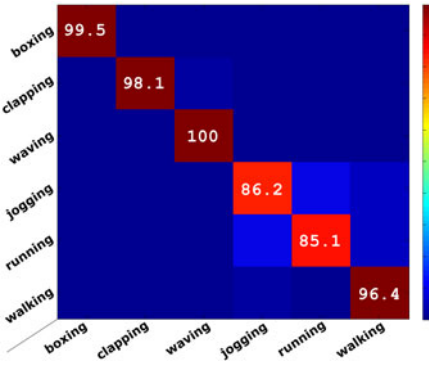


Fig. 6. Our confusion matrix for KTH

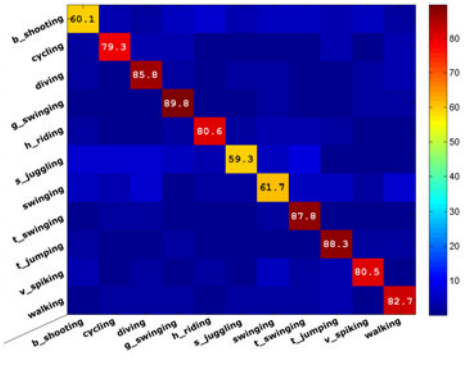


Fig. 7. Our confusion matrix for UCF videos

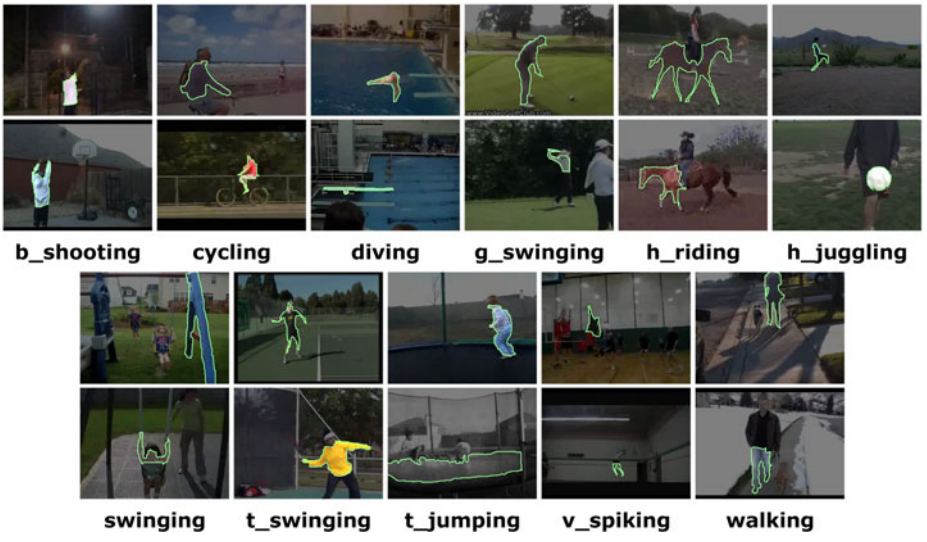


Fig. 8. Examples of the learned codewords from the UCF "YouTube" dataset. The codewords are highlighted in the frames of exemplar videos from which the codewords have been extracted.

from the Weizmann dataset, and take queries from the CMU "Crowded" videos. Table 2 shows the area under the ROC curve (AUC) that we have obtained for LP-ℓ_1 by varying the values of input parameters α and β. As can be seen, even when our training occurs on the sanitized dataset, our AUC values, for four different activity classes, are better than that of the competing approaches [3, 8].

Other Evaluation: Table 5 shows that we compare favorably with the state-of-the-art. We also provide confusion matrices for KTH and UCF "YouTube" datasets in Fig. 5 and Fig. 5. Fig. 8 shows two examples of the learned codewords for each class of the "YouTube" dataset. As can be seen, the codewords may represent only a body part, or objects defining the activity (the trampoline for 't_jumping' or the swinging gear for 'swinging').

6 Conclusion

We have shown that certain human actions can be efficiently represented by short time series of activity codewords. The codewords represent still snapshots of human body parts in their discriminative postures, relative to other activity classes. In addition, the codewords may represent discriminative objects that people interact with while performing the activity. Typically, our time series representation compresses the original hundreds of video frames to only about 10 key human postures. This carries many advantages for developing a robust, efficient, and scalable activity recognition system. Our main focus has been on specifying four alternative methods for learning the dictionary of codewords from a large set of static and local-motion video features, under only weak

supervision. We have formulated this learning as maximization of the hypothesis margin of the 1-NN classifier with ℓ_1 and ℓ_2 regularization. For the four learning methods, we have presented strong theoretical guarantees of their convergence to the globally optimum solution. The methods have linear complexity in the number of video features, and small generalization error. We have evaluated the proposed time-series representation on the challenging problem of activity detection and localization in realistic videos (YouTube) with dynamic, cluttered backgrounds. Our activity recognition yields better performance when using a set of dictionaries learned per each activity class than the all-class dictionary. Interestingly, significant classification-accuracy gains are achieved when using the short time series of codewords vs. a long sequence of codewords (one per each video frame) as the video representation. Our results show that, with small computation times, we outperform the state of the art on the benchmark datasets.

References

1. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
2. Niebles, J.C., Han, B., Ferencz, A., Fei-Fei, L.: Extracting moving people from internet videos. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 527–540. Springer, Heidelberg (2008)
3. Yao, B., Zhu, S.C.: Learning deformable action templates from cluttered videos. In: ICCV (2009)
4. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos "in the wild". In: CVPR (2009)
5. Fanti, C., Zelnik-Manor, L., Perona, P.: Hybrid models for human motion recognition. In: CVPR (2005)
6. Bissacco, A., Yang, M.H., Soatto, S.: Detecting humans via their pose. In: NIPS (2007)
7. Ning, H., Xu, W., Gong, Y., Huang, T.: Discriminative learning of visual words for 3d human pose estimation. In: CVPR (2008)
8. Ke, Y., Sukthankar, R., Hebert, M.: Event detection in crowded videos. In: ICCV (2007)
9. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In. CVPR, pp. 886–893 (2005)
10. Lin, Z., Davis, L.S.: A pose-invariant descriptor for human detection and segmentation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 423–436. Springer, Heidelberg (2008)
11. Marszalek, M., Laptev, I., Schmid, C.: Actions in context. In: CVPR (2009)
12. Lin, Z., Jiang, Z., Davis, L.S.: Recognizing actions by shape-motion prototype trees. In: ICCV (2009)
13. Liu, J., Shah, M.: Learning human actions via information maximization. In: CVPR (2008)
14. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. IEEE TPAMI 29, 2247–2253 (2007)
15. Brendel, W., Todorovic, S.: Video object segmentation by tracking regions. In: ICCV (2009)
16. Frome, A., Singer, Y., Sha, F., Malik, J.: Learning globally-consistent local distance functions for shape-based image retrieval and classification. In: ICCV (2007)
17. Gilad-Bachrach, R., Navot, A., Tishby, N.: Margin based feature selection – theory and algorithms. In: ICML, vol. 43 (2004)
18. Vedaldi, A., Soatto, S.: Quick shift and kernel methods for mode seeking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 705–718. Springer, Heidelberg (2008)
19. Schueldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local SVM approach. In: ICPR (2004)

Fast Approximate Nearest Neighbor Methods for Non-Euclidean Manifolds with Applications to Human Activity Analysis in Videos

Rizwan Chaudhry[1,⋆] and Yuri Ivanov[2]

[1] Center for Imaging Science, Johns Hopkins University
3400 N Charles St, Baltimore, MD 21218, USA
rizwanch@cis.jhu.edu
[2] Mitsubishi Electric Research Laboratories
201 Broadway, Cambridge, MA 02139, USA
yivanov@merl.com

Abstract. Approximate Nearest Neighbor (ANN) methods such as Locality Sensitive Hashing, Semantic Hashing, and Spectral Hashing, provide computationally efficient procedures for finding objects similar to a query object in large datasets. These methods have been successfully applied to search web-scale datasets that can contain millions of images. Unfortunately, the key assumption in these procedures is that objects in the dataset lie in a Euclidean space. This assumption is not always valid and poses a challenge for several computer vision applications where data commonly lies in complex non-Euclidean manifolds. In particular, dynamic data such as human activities are commonly represented as distributions over bags of video words or as dynamical systems. In this paper, we propose two new algorithms that extend Spectral Hashing to non-Euclidean spaces. The first method considers the Riemannian geometry of the manifold and performs Spectral Hashing in the tangent space of the manifold at several points. The second method divides the data into subsets and takes advantage of the *kernel trick* to perform non-Euclidean Spectral Hashing. For a data set of N samples the proposed methods are able to retrieve similar objects in as low as $O(K)$ time complexity, where K is the number of clusters in the data. Since $K \ll N$, our methods are extremely efficient. We test and evaluate our methods on synthetic data generated from the Unit Hypersphere and the Grassmann manifold. Finally, we show promising results on a human action database.

Keywords: Approximate Nearest Neighbors, Hashing, Non-Euclidean Manifolds, Activity Analysis in Videos.

1 Introduction

Human action analysis is considered one of the most important problems in computer vision. It enables such applications as automatic surveillance, behavior

⋆ This work was done as part of a summer research internship at the Mitsubishi Electric Research Laboratories, Cambridge, MA, USA.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 735–748, 2010.

analysis, elderly care, etc. There has been a tremendous amount of work towards automatic analysis of human motion in videos. However, due to extensive amount of computation required for video analysis, this work by necessity is often restricted to smaller models and datasets. In a real-life surveillance scenario video data is continuously recorded for a long period of time and saved for later analysis. Search in such extensive volumes of data remains a difficult task. Toward this goal, this paper proposes a major step in developing hashing techniques upon which sophisticated and efficient searches for a nearest neighbor in large corpora of video data can be built. It is often the case when classifying complex data, that using sophisticated features makes even a simple NN technique perform very well. Sampling from a neighborhood, commonly used in tracking applications, can also benefit from efficiency of hashing-based NN search. In this work we present two methods that have a goal of replicating the Nearest Neighbor search to make it applicable to very large datasets of complex features.

Prior work. Recently, there has been a surge in interest in fast content-based image retrieval from web-scale databases of tens of millions of images. However there has been little work in the same direction for videos. In this section we give a summary of notable work dominant in the field. Karpenko *et. al.* in [1] introduced a method where all the videos in a dataset were compressed to very small frame sizes and only a few key-frames. Using intensity statistics in the frames, the comparison of the new query is performed with the entire dataset. This technique leads to faster video comparison, but doesn't use semantically meaningful features and cannot be performed faster than $O(N)$. Biswas *et. al.* [2] provided a method that used two-level hash tables based on the invariant geometric properties of object shapes for efficient search and retrieval. Turaga *et. al.* [3] proposed a dynamical-systems based model for human activities that can be used for clustering different types of activities in a continuous video. Sidenbladh *et. al.* [4] used an approximate probabilistic tree search to find the closest match in the database for a query motion. Ben-Arie *et. al.* in [5] used a sparsely sampled sequence of body poses and velocity vectors of body parts as they move in a scene to construct multi-dimensional hash tables. For a test video, these features were extracted and the key was used to find the match in the hash-tables. Several other methods proposed in [6,7,8] cluster features derived from motion and appearance information for semantic retrieval.

All of these methods either use exact-match hashing, which generally has difficulties in performing a neighborhood search, or tree-based approaches, which often help increase performance, but are not as fast as hashing techniques.

Recently, new hashing algorithms that preserve neighborhood relationship between derived codes have been developed. Approximate Nearest-Neighbor methods such as the variants of Locality Sensitive Hashing (LSH), [9], Semantic Hashing, [10], and Spectral Hashing, [11], provide efficient algorithms for constructing binary codes for points in a high dimensional space. These methods have the property that codes for points that are nearby in the high-dimensional space are also close to each other in the binary code space under the Hamming distance. This provides an excellent method for creating hash tables because even if the

key for a query object is not in the table, the keys for neighbors in the Hamming space can then be checked by simply flipping a bit of the binary code.

One important limitation of all the above methods is that they are only applicable to data that resides in a Euclidean space. However, features frequently used for activity analysis in dynamic data have strong non-Euclidean character. For instance, histograms created as part of a bags of video words classification procedure on local features proposed by Laptev, [12] and Dollar et al. [13], or dynamical systems proposed in [14,15,16,17], naturally lie on a non-trivial manifold that has strong non-Euclidean properties. Hence the above methods are not directly applicable. The authors in [11] mentioned this limitation of Spectral Hashing and assumed that a suitable Euclidean embedding can be used. However, finding such an embedding is not always possible. A workaround for LSH that uses the *kernel trick* to implicitly embed the data in a high-dimensional Euclidean space is proposed by Kulis and Grauman [18]. Further, Kulis and Darrell in [19] use a similar kernel trick for Spectral Hashing. However as we will explain later, this method is no faster than performing exact nearest neighbor search.

As shown in [11], LSH usually gives very large codewords, whereas Semantic Hashing and Spectral Hashing give compact binary codewords and therefore are more useful for mapping objects directly to memory addresses in a computer. In this paper, we turn our attention to Spectral Hashing and propose two new fast approximate methods for performing Spectral Hashing on non-Euclidean data. In section 2 we summarize standard Euclidean spectral hashing and formulate the exact problem for non-Euclidean data. In section 3, we explain our proposed methods and their complexity. In section 4 we test our algorithm on both synthetic and real data sets; and give future directions of research in section 5.

2 Spectral Hashing

As presented by Weiss et al. in [11], given data points, $\{\mathbf{x}_i\}_{i=1}^N \in \mathbb{R}^d$, the goal is to find k-bit binary vectors, $\{y_i\}_{i=1}^N \in \{-1,1\}^k$ such that similar points in $\mathbb{R}^N$, under the similarity measure, $W_{ij} = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\epsilon^2})$ map to binary vectors that are close to each other under the Hamming distance weighted by W.

If we assume that the data, $\mathbf{x}_i \in \mathbb{R}^d$, is sampled from a probability distribution $p(\mathbf{x})$, Spectral Hashing (SH) solves the following optimization problem:

$$\text{minimize} \quad \int \|y(\mathbf{x}_1) - y(\mathbf{x}_2)\|^2 W(\mathbf{x}_1, \mathbf{x}_2) p(\mathbf{x}_1) p(\mathbf{x}_2) d\mathbf{x}_1 d\mathbf{x}_2 \tag{1}$$

$$\text{s.t.} \quad y(\mathbf{x}) \in \{-1,1\}^k, \quad \int y(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = 0, \quad \text{and} \quad \int y(\mathbf{x}) y(\mathbf{x})^\top p(\mathbf{x}) d\mathbf{x} = I$$

Relaxing the first constraint gives the solution of the problem, y as the first k *eigenfunctions* of the weighted Laplace-Beltrami operator on the manifold. If p is the multi-dimensional uniform distribution on a subset of $\mathbb{R}^d$ and the weighting function, W, is defined as above, there exists a one-shot closed form solution for these eigenfunctions. However, in the case of a Gaussian distribution on $\mathbb{R}^d$, there exists an iterative solution.

Spectral hashing has a very appealing mathematical formulation. Ideally, one could take any probability distribution on a general manifold and a weighting function and analytically compute the eigenfunctions of the corresponding Laplace-Beltrami operator. However, even in the simpler case of Euclidean data, a closed form solution might not exist. Thus, analysis of non-Euclidean data may require solving this problem numerically. Furthermore, the weighting function, W, is computed from geodesic distances and thus, is no longer a simple exponential similarity. This makes the exact computation of the solution of the minimization problem in Eq. (1) computationally intractable.

As part of computing the eigenfunctions, the method in [11] employs PCA to compute a basis for the dataset, and then computes closed-form one-dimensional eigenfunctions of the weighted Laplace-Beltrami operator in each principal component direction using a rectangular approximation to the spread of the data. The method then combines these 1-D eigenfunctions to compute the eigenfunctions of the original dataset in $\mathbb{R}^d$. To deal with non-Euclidean data, Kulis *et. al.* [19] proposed using Kernel PCA instead of PCA in this step. We will refer to this method as *Kernel Spectral Hashing* (KSH). Even though their method is theoretically correct, as the kernels would embed the points in a high-dimensional Euclidean space, finding the value of the eigenfunction at each new test datapoint would involve computing the kernel of the test point with all the points in the training set used to compute the kernel PCA components. Because of this, even though a well-chosen kernel might give fine retrieval accuracy, the computational complexity of this method is at least $O(N)$.

3 Non-Euclidean Spectral Hashing

Noting the difficulty with applying Spectral Hashing techniques to non-Euclidean manifolds, we propose two new methods for finding compact binary codes for data lying on such manifolds with which this difficulty can be circumvented.

3.1 Riemannian Spectral Hashing

Since it is hard to compute closed form eigenfunctions in the SH algorithm for non-Euclidean data, we can embed the data in a Euclidean space. Then, under the assumption that it is drawn from a uniform distribution in that space, spectral hashing can be applied in this embedding space. Our first method, *Riemannian Spectral Hashing (RSH)*, follows this strategy.

The tangent space, $\mathcal{T}_{\mathbf{y}}\mathcal{M}$, to a manifold, $\mathcal{M}$ at a point $\mathbf{y}$ is a Euclidean space. Therefore, assuming that the manifold is geodesically complete, the data, $\{\mathbf{x}_i\}_{i=1}^N$, can be projected onto the tangent space at $\mathbf{y}$ by using the *logarithm map*, $\Delta_i = \overrightarrow{\mathbf{yx}_i} = \log_{\mathbf{y}}(\mathbf{x}_i)$. This makes it possible to perform Spectral Hashing on the tangent space projections, $\{\Delta_i\}_{i=1}^N$ locally, around $\mathbf{y}$ without introducing significant projection error. In order to minimize projection errors, the RSH algrorithm approximates a manifold with a *set of tangent hyperplanes*, positioned on a set of representative points (poles) which follow the distribution of the data

on the manifold. The poles are found by clustering, for which we can use any extrinsic manifold clustering algorithm such as [20], [21] to cluster the data on the manifold into K clusters. We use the Riemannian k-means procedure:

1. Initialize cluster centers, $\{\mathbf{c}_j\}_{j=1}^{K}$ by randomly choosing K points from the training data.
2. For each point $\mathbf{x}_i$ in the data set, compute the geodesic distance to each cluster center, $d(\mathbf{c}_j, \mathbf{x}_i) = \| \log_{\mathbf{c}_j}(\mathbf{x}_i) \|$. Assign the cluster center that is the closest to the data point as the cluster membership, $w_i = \text{argmin}_j \| \log_{\mathbf{c}_j}(\mathbf{x}_i) \|$.
3. Recompute each cluster center as the Karcher mean of the points in each cluster, $\mathbf{c}_j = \text{mean}\{\mathbf{x}_l | w_l = j\}$. This requires repeated uses of the *exponential map* and the *logarithm map* on the manifold until convergence to a mean.
4. Repeat until convergence.

This clustering algorithm is a simple extension of the k-means algorithm to a single geodesically complete manifold under the assumption that no two points in the same cluster are antipodes. This method inherits the convergence properties of regular Euclidean k-means. Once the clusters, $\{\mathbf{c}_j\}_{j=1}^{K}$, and memberships, $\{w_i\}_{i=1}^{N}$, have been assigned, all the points in the same cluster are projected to the tangent space around the cluster center using the corresponding logarithm maps. A separate spectral hashing algorithm is then trained on each tangent space.

For computing the binary code of a new test point, $\mathbf{z}$, we first compute the geodesic distance of $\mathbf{z}$ with all the cluster centers and project it to the tangent space of the closest cluster center, $\mathbf{c}_k$, where $k = \arg\min_j \|\overrightarrow{\mathbf{c}_j\mathbf{z}}\|$ to get $\Delta_{\mathbf{z}} = \log_{\mathbf{c}_k}(\mathbf{z})$. We then use spectral hashing to find the binary code of $\Delta_{\mathbf{z}}$. Since finding the right cluster center, only requires K geodesic distance evaluations, this results in a computational cost of $O(K)$. Even though this is greater than $O(1)$ as in Spectral Hashing, it is much less than $O(N)$ as in Kernel Spectral Hashing, where $K \ll N$. Moreover, by clustering all the data, we better approximate the uniform distribution assumption in each cluster. We summarize RSH in Algorithm 1. Figure 1(b) provides an illustration of the method.

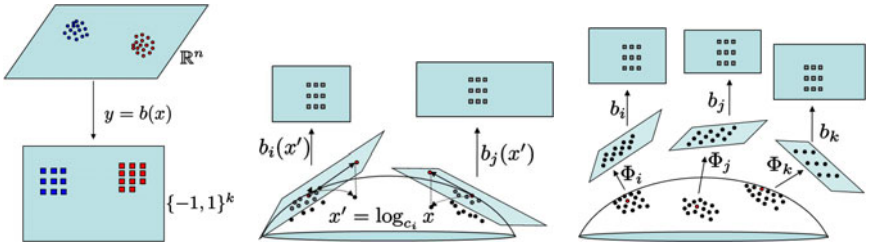
Algorithm 1: Riemannian Spectral Hashing

Training

1. Cluster training data using a manifold clustering algorithm, [20], [21].
2. Compute log-maps and project each cluster to the tangent space around center.
3. Train spectral hashing on points in each tangent space separately.

Testing

1. Find closest cluster center using geodesic distances on the manifold.
2. Project onto tangent space around closest cluster center.
3. Compute binary code of the projected point using Spectral Hashing.
4. Retrieve the nearest neighbor.



(a) Spectral Hashing (SH) [11] (b) Riemannian Spectral Hashing (RSH) (c) Distributed Kernel Spectral Hashing (DKSH)

Fig. 1. Schematic diagram of state-of-the-art and proposed hashing methods

3.2 Distributed Kernel Spectral Hashing (DKSH)

In certain cases, closed form experssions for the logarithm and exponential maps for manifolds are not available. This limits the applicability of extrinsic manifold clustering algorithms as required in Alg. 1. If however, a kernel or other affinity measure, $W(.,.)$ is defined on the manifold, a non-linear dimensionality reduction method such as Multidimensional Scaling (MDS) [22] can be employed to project the data into a low-dimensional Euclidean space before performing k-means on this low-dimensional space. Alternatively, a non-linear clustering algorithm such as kernel k-means [23] or Spectral Clustering [24] can be used to compute cluster associations of the data. As a result, we would not have cluster centers but only cluster associations for the training data. After the clustering stage, one representative point is chosen in each cluster to represent all data within it. One method to choose this point is as follows [25]:

1. Compute the $N \times N$ affinitiy matrix, W, of the training data based on a kernel or affinity defined on the manifold.
2. Perform MDS using W to get a low-dimensional Euclidean representation $\{\mathbf{u_i}\}_{i=1}^{N}$ and perform k-means on these points to get K cluster centers $\{\mathbf{v}_j\}_{j=1}^{K}$ in the low-dimensional space.
3. Within each cluster center, choose the point $\mathbf{u}$ in the projected data that is closest to each cluster center $\mathbf{v}_j$.
4. Find the original points $\{\mathbf{x}_{p;j}\}_{j=1}^{K}$ on the manifold that mapped to the points $\{\mathbf{v}_j\}_{j=1}^{K}$ after MDS and use these points as cluster representatives (pivots).

Once a representative, or pivot, for each cluster has been computed, we train Kernel Spectral Hashing (KSH) separately for each cluster.

As in RSH, to find the binary code for a test point, $\mathbf{z}$, we first compute its affinity, $W(\mathbf{x}_{p;j}, \mathbf{z})$ with each pivot point and assign $\mathbf{z}$ to the j-th cluster if $\mathbf{x}_{p;j}$ has the highest affinity with $\mathbf{z}$. We then use Kernel Spectral Hashing trained for that specific cluster and compute the binary code for $\mathbf{z}$ to retrieve the nearest neighbors. Assuming that in the best case, all the points are equally divided between K clusters, the query time complexity of this method is $O(K + N/K)$ on average, which is more computationally expensive that RSH. However, it is

still significantly better than the complexity of KSH. In the worst case when only 1 cluster is chosen, the complexity is $O(N)$, the same as KSH. We summarize DKSH in Algorithm 2. Figure 1(c) provides an illustration of the method.

Algorithm 2: Distributed Kernel Spectral Hashing

Training

1. Cluster training data using Non-Linear clustering (MDS, Spectral clustering etc.) using kernel similarity
2. Pick a pivot point, representing each cluster.
3. Train Kernel Spectral Hashing on points in each cluster separately.

Testing

1. Use kernel similarity to compute pivot with the highest affinity to test point.
2. Compute binary code with kernel spectral hashing for that pivot.
3. Retrieve the nearest neighbor.

4 Experiments

In this section we compare the proposed methods, Riemannian Spectral Hashing (RSH) and Distributed Kernel Spectral Hashing (DKSH), against exact Nearest Neihbors (NN), and state-of-the-art Hashing methods: Kernel Locality Sensitive Hashing (KLSH) [18], Euclidean Spectral Hashing [11] (SH), and Kernel Spectral Hashing [19] (KSH).

4.1 Synthetic Data

We first test the proposed methods on synthetic datasets of points lying on two non-Euclidean manifolds: the 100 dimensional unit hypersphere, S^{99}, and the manifold of all 3-dimensional subspaces of $\mathbb{R}^{10}$, *i.e.*, the Grassmann manifold, $G_{10,3}$ or $G_{3,10-3}$. The evaluation is performed on an 8-core Intel Xeon 3.4 GHz machine with 32 GB of RAM. In each experiment, we restrict the number of processing cores to exactly one so that the run-times of various algorithms are comparable. When comparing our methods with Spectral Hashing, we treat the points on both the above mentioned manifolds as points in $\mathbb{R}^{100}$ and $\mathbb{R}^{30}$ respectively.

As a technical detail, it is noted that when the data size grows larger than 10^4 samples, the memory requirements of the PCA computation in SH and the kernel PCA in KSH become extremely large and can not be handled by our computational resources. As an example, consider computing the all pair kernel matrix for 10^5 points. Storing the result as a double precision matrix in memory requires a minimum of $(10^5)^2 \times 8 = 72$ GB of memory, which is not available in our system. Therefore for datasets larger than 10^4 points, we randomly sample 1000 points, equally sampling from each class, and pre-train all the hashing algorithms on this smaller set. We then compute the binary hash codes for all

the training points and store them for comparison against the test sets. Since the number of exponential and logarithm map evaluations as well as kernel evaluations will decrease, we will distinguish the training and testing times for the hashing methods where all the data was used for training ($10-10^4$ samples), and for methods where a pre-training approach was used, ($10^5, 10^6$ samples).

Unit hypersphere - S^{99}. The unit hypersphere, S^{99}, is the set of all points, $\mathbf{x} \in \mathbb{R}^{100}$ that satisfy the constraint, $\sum_{i=1}^{100} x_i^2 = 1$. The geodesic distance between two points, $\mathbf{x}$ and $\mathbf{y}$, on a hypersphere is defined as $d_G(\mathbf{x}, \mathbf{y}) = \cos^{-1}(\mathbf{x}^\top \mathbf{y})$. Moreover, the logarithm and exponential maps on the sphere are defined as,

$$\log_{\mathbf{x}}(\mathbf{y}) = \frac{\mathbf{y} - (\mathbf{x}^\top \mathbf{y})\mathbf{x}}{\|\mathbf{y} - (\mathbf{x}^\top \mathbf{y})\mathbf{x}\|} \cos^{-1}(\mathbf{x}^\top \mathbf{y}),$$

$$\exp_{\mathbf{x}}(\Delta) = \cos(\|\Delta\|)\mathbf{x} + \sin(\|\Delta\|)\frac{\Delta}{\|\Delta\|},$$

where Δ is a tangent vector at the pole $\mathbf{x}$. Finally, the standard inner product also defines a kernel on the sphere, i.e. $k(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y}$.

We generate 5 sets of 5-class each training datasets containing 100, 1000, 10^4, 10^5 and 10^6 points on S^{99}. For testing, we generate 100 more points in each case. Figure 2 displays the difference between the recognition percentages of exact 1-NN and the state-of-the-art methods (Kernel LSH (KLSH), SH and KSH) and the proposed hashing algorithms (RSH and DKSH). We use 8 bits for all hashing algorithms and 5 clusters for the proposed methods. Both RSH and DKSH have the lowest percentage difference compared to the state-of-the-art methods for all training sizes. Moreover, the error percentages remain within 10-15% of the exact 1-NN method. This can clearly be attributed to the fact that the proposed methods specifically take into account the manifold structure of the space and thus result in better recognition performance.

Table 1 shows the training times required for each algorithm against the number of training samples. 1-NN does not require any training, whereas SH and

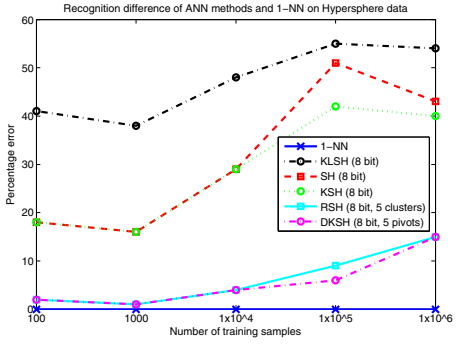


Fig. 2. S^{99} - Comparison of NN and ANN methods

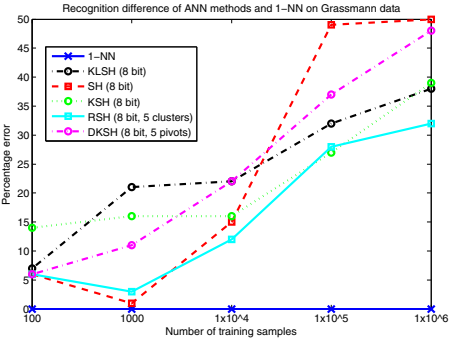


Fig. 3. $G_{10,3}$ - Comparison of NN and ANN methods

Table 1. S^{99} - Training times

Method	Training time (sec)				
# Training	100	1000	10^4	10^5	10^6
NN	0	0	0	0	0
KLSH	0.01	3.44	1.5h	85.0	11.7m
SH	0.01	0.40	56.02	46.11	6.4m
KSH	0.02	7.30	2.0h	50.7m	7.5h
RSH	0.35	1.90	33.8	70.17	8.8m
DKSH	0.08	7.62	2.0h	72.02	10.1m

Table 2. S^{99} - Testing times

Method	Testing time (sec)				
# Training	100	1000	10^4	10^5	10^6
NN	0.01	0.02	0.41	4.81	1.3m
KLSH	0.03	0.02	0.37	1.96	17.0
SH	0.04	0.04	0.04	0.09	1.06
KSH	0.06	3.13	4.1m	2.32	4.02
RSH	0.06	0.06	0.05	0.07	0.28
DKSH	0.06	0.07	10.09	0.07	0.25

Table 3. $G_{10,3}$ - Training times

Method	Training time (sec)				
# Training	100	1000	10^4	10^5	10^6
NN	0	0	0	0	0
KLSH	0.01	5.07	1.2h	34.0m	5.5h
SH	0.01	0.08	9.60	16.63	3.9m
KSH	0.12	17.2	2.5h	1.5h	13.2h
RSH	0.56	10.52	6.1m	10.2m	16.6m
DKSH	0.17	21.27	3.2h	36m	6.3h

Table 4. $G_{10,3}$ - Testing times

Method	Testing time (sec)				
# Training	100	1000	10^4	10^5	10^6
NN	2.06	21.2	3.3m	41.1m	5.7h
KLSH	0.20	2.29	22.5	4.15	23.1
SH	0.04	0.03	0.04	0.10	1.27
KSH	0.28	3.60	3.5m	4.83	5.47
RSH	0.12	0.11	0.10	0.13	1.21
DKSH	0.17	1.95	1.5m	1.16	1.88

RSH are the fastest to train. The training times for KLSH, KSH and DKSH increase greatly with the number of training samples. Table 2 provides the total test time for 100 samples. Coupled with higher accuracy, this is where we observe the real advantage of the proposed methods. As the size of the training data increases, not surprisingly, the time taken for 1-NN also increases. All test times for SH and KSH remain low but are still higher than the test times for RSH and DKSH. This again illustrates the superiority of the proposed methods.

Finally, Figure 4 displays the dependence of the *error rate* of RSH on the algorithm parameters, *i.e.*, the number of bits and the number of clusters. We can see that if the number of bits is kept constant, increasing the number of cluster centers decreases the testing error rate. Similarly, keeping the number of clusters constant, and increasing the number of bits also decreases the testing error rate. The first quality is highly desirable, since in a real scenario, the binary code will represent the memory location for a pointer to the data. Thus having more than 64 bits is not practical. In fact this shows that we can use relatively fewer number of bits and pack the data points in memory by using more clusters. Since the clusters can be located arbitrarily in memory, this reduces the need for large chunks of contiguous memory.

Grassmann manifold - $G_{10,3}$. In an analogous fashion to the previous section, we generate several training samples of different sizes on the Grassmann manifold, $G_{10,3}$, which is the manifold of all the 3-dimensional subspaces of $\mathbb{R}^{10}$. The data lies in 5 classes and is generated using the method in [26]. For non-linear clustering and tangent space to manifold projections and vice-versa, we use the expressions for the exponential and logarithm maps on the Grassmann manifold

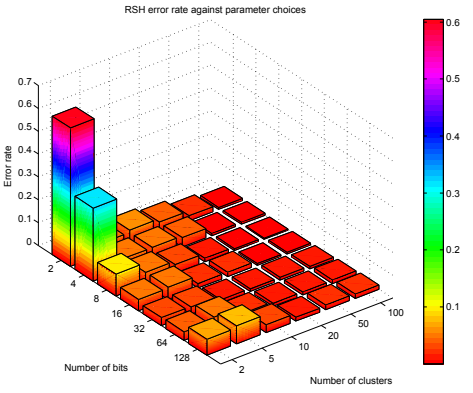


Fig. 4. S^{99} - RSH error rates against algorithm parameters

in [20]. For computing the kernel on the manifold we use the product of the cosines of the subspace angles between the subspaces [27]. Again, we use 8 bits for the binary codes for all hashing algorithms and 5 clusters for our proposed methods. Figure 3 displays the difference between the recognition percentage of 1-NN and all other methods. At first it might seem that SH performs better than the other methods for small data sizes, the trend is offset drastically with large training sizes where it performs the worst. Overall, RSH performs better than all state-of-the-art methods and the error stays within 30% of that of 1-NN.

Table 3 provides the training times for each of the training datasets. We notice that KLSH, KSH and DKSH require the largest training times, whereas RSH and SH require the least. For all the methods, the training time increases with the number of data points but due to the large number of kernel computations during the training stage, the increase in time is greatest for KLSH, KSH and DKSH. Table 4 gives the total test time for 100 test samples for each of the training sizes. We again see the computational advantage of the proposed methods against the exact method as well as the state of the art KLSH and KSH methods. The test time increases steeply with the size of the data for the kernel-based methods, whereas the corresponding increase in test time stays low for RSH.

From the above set of experiments, we have shown that the proposed approximate nearest-neighbor methods, RSH and DKSH, by explicitly considering the manifold structure of the space of data, provide great computational advantage against exact Nearest Neighbors while having very low to modest decrease in accuracy. Moreover, our methods always perform better than KLSH and KSH, the state-of-the-art non-Euclidean Hashing methods.

4.2 Human Action Dataset

Recent approaches in human activity recognition use features such as (1) distributions over a bag of spatial-temporal keypoints to represent the activity in a scene, or (2) dynamical systems learnt from a time-series of features extracted from the frames of the video. Both these features lie in non-Euclidean spaces

and therefore the proposed approach is directly applicable for the purpose of retrieving activities from a large dataset of activity videos. Even though, human activity analysis has been a vibrant field in computer vision, to the best of our knowledge, no datasets are available that contain more than a few thousand instances of human actions. Videos of unstructured scenes with multiple activities and events are available, however, the ground-truth activity segmentation and tracking is not provided and automatic extraction of these remains an open problem in computer vision. One of the most popular and largest datasets available is the KTH human action dataset [28]. This dataset contains six actions: walking, running, jogging, boxing, handwaving and handclapping. There are 25 persons performing these actions under four different scenarios: outdoors, outdoors across different scales, outdoors with bulky clothes on and indoors. There are a total of 2391 sequences in the dataset.

For our first experiment, we use the approach of [13] and extract several spatio-temporal keypoints and their corresponding descriptors in all the videos. We divide the data as follows: All the videos of the first 16 subjects are used for training whereas the videos of the remaining 9 subjects are used for testing. A k-means procedure is used to cluster the descriptors in the training data to form a dictionary of 100 keypoints. We then learn feature distributions for each action video around these keypoints. This provides a 100 dimensional histogram per video that represents the action in that video. These histograms are used for training and testing the proposed ANN methods. Note that for a fair comparison to nearest-neighbor algorithms, we will test our method against the simple nearest-neighbor algorithm and not against the state-of-the-art methods for human activity recognition that use sophisticated classification algorithms to achieve superior performance. The error rates reported below are not state-of-the-art on the KTH human action database; instead, they are the error rates achieved when using exact NN and state-of-the-art ANN methods and our proposed methods on the dataset. We emphasize that our goal here is not to find the best classification algorithm on the KTH database, but to compare the performance of the proposed ANN methods against state-of-the-art ANN and exact NN methods. Table 5 compares the performance of the proposed methods with Nearest Neighbors and state-of-the-art hashing methods. All the hashing methods use 8-bits for the binary codes. The proposed methods, RSH and DKSH divide the training data into 3 clusters. The results show that RSH has the highest recognition percentage other than exact NN, whereas the state-of-the-art KLSH has the worst recognition percentage. Moreover, RSH is also the most efficient method in terms of retrieval time, even though it requires the largest training time. Furthermore, the best recognition rate achievable using RSH was 69% with 64 bit code-words and 2 clusters, which is only 7% below the error rate achieved by exact 1-NN.

For our second experiment, we use the approach in [17] and compute the Histogram of Oriented Optical Flow (HOOF) features at each frame to get a normalized histogram time-series for each video. We then learn a linear-state nonlinear dynamical system (NLDS) using the approach in [17] with the Geodesic

Table 5. BOW histograms

Method	Correct %	Train t	Test t
NN	76	0	11.5
KLSH	24	38.4	1.04
SH	51	3.4	0.45
KSH	51	41.1	81.1
RSH	62	58.6	0.39
DKSH	51	31.1	3.34

Table 6. Observability matrices

Method	Correct %	Train t	Test t
NN	72	0	149.3
KLSH	64	0.547	35.0
SH	22	5.262	1.67
KSH	17	31.24	38.7
RSH	65	321.7	10.3
DKSH	58	266.5	15.8

(Bhattacharya) kernel on histograms. Hence each activity video is now represented as a non-linear dynamical system. There are several methods for comparing dynamical systems, *e.g.* those proposed in [17] and the references therein. We represent the dynamics and output transformation functions using the observability matrix for each dynamical system. Since we are using the inner-product on the sphere as the kernel, we can simply use PCA to learn the approximate dynamical system parameters and thus get the parameter matrices, $A \in \mathbb{R}^{n \times n}$ and $C \in \mathbb{R}^{p \times n}$. Here n is the system order, and p is the size of the output, 10 and 64, respectively, in our case. The observability matrix can then be computed as $\mathcal{O} = [C^\top, (AC)^\top (A^2C)^\top, \ldots, (A^{n-1}C)^\top]^\top \in \mathbb{R}^{pn \times n}$. See [17] and the references therein for more details. Notice that the columns of $\mathcal{O}$ span an n-dimensional subspace of $\mathbb{R}^{np}$ and thus $\mathcal{O}$ lies on the Grassmann manifold, $G_{np,n}$. We can therefore follow the experiments in section 4.1 using these observability matrices as the data points.

Since the approach in [17] is directly applicable only for stationary cameras, we choose sequences from the first scenario, i.e. outdoors with stationary camera (around 600 sequences) to test our algorithms. We use 64% of the data for training and the remaining 36% for testing. Moreover, we use 64 bits for the binary codes and 15 clusters/pivots for the proposed methods.

Table 6 shows the recognition percentages and training and testing times for exact KNN using the Martin distance for dynamical systems, and the proposed and state-of-the-art hashing methods. We can see that our method, RSH, has the best recognition rate, slightly above KLSH. Notice that even though exact NN does not require any training, which could be as high as 321.7 seconds for RSH, the speed up in terms of test times is significant. Exact KNN requires 149.3 seconds for testing whereas RSH requires only 10.3 seconds and DKSH requires 15.8 seconds. Even though KLSH performs well in this scenario, due to the many kernel computations required, its testing time is at least 3 times greater than RSH, limiting the former's advantage.

5 Conclusion and Future Work

We have proposed two new methods, Riemannian Spectral Hashing (RSH), and Distributed Kernel Spectral Hashing (DKSH), for performing fast approximate

nearest-neighbor matching on non-Euclidean data. We have shown that state-of-the-art methods either do not take into account the manifold structure of the data, or are computationally inefficient and can in fact be slower in performance than exact nearest neighbors. Moreover, experiments on synthetic and real data have shown that our methods are applicable to points that lie on simple manifolds such as the unit hypersphere as well as to points that lie on highly complicated manifolds such as the space of dynamical systems. The proposed methods provide immense computational savings at the cost of a small decrease in accuracy and hence are ideal for approximate nearest neighbor matching in large datasets. We have provided average-case time complexity for our proposed methods and are looking into how the parameters such as the number of bits and number of clusters/pivots, can be set so as to achieve user-defined precision/recall tolerances. Finally we are working on collecting a very large human action dataset to further validate the benefits of our proposed methods.

Acknowledgments. The authors would like to thank Kinh Tieu, Ashok Veeraraghavan and Oncel Tuzel for their comments and discussions that helped improve the presentation of this work.

References

1. Karpenko, A., Aarabi, P.: Tiny videos: Non-parametric content-based video retrieval and recognition. In: IEEE International Symposium on Multimedia (2008)
2. Biswas, S., Aggarwal, G., Chellappa, R.: Efficient indexing for articulation invariant shape matching and retrieval. In: IEEE Conference on Computer Vision and Pattern Recognition (2007)
3. Turaga, P., Veeraraghavan, A., Chellappa, R.: From videos to verbs: Mining videos for events using a cascade of dynamical systems. In: IEEE Conference on Computer Vision and Pattern Recognition (2007)
4. Sidenbladh, H., Black, M.J., Sigal, L.: Implicit probabilistic models of human motion for synthesis and tracking. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 784–800. Springer, Heidelberg (2002)
5. Ben-Arie, J., Wang, Z., Pandit, P., Rajaram, S.: Human activity recognition using multidimensional indexing. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 1091–1104 (2002)
6. Chen, D.Y., Lee, S.Y., Chen, H.T.: Motion activity based semantic video similarity retrieval. In: Advances in Multimedia Information Processing (2002)
7. Chen, X., Zhang, C.: Semantic event retrieval from surveillance video databases. In: IEEE International Symposium on Multimedia (2008)
8. Kashino, K., Kimura, A., Kurozumi, T.: A quick video search method based on local and global feature clustering. In: International Conference on Pattern Recognition (2004)
9. Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In: Symposium on Foundations of Computer Science (2006)
10. Salakhutdinov, R., Hinton, G.: Semantic hashing. International Journal of Approximate Reasoning 50, 969–978 (2009)

11. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: Neural Information Processing Systems Conference (2008)
12. Laptev, I.: On space-time interest points. International Journal of Computer Vision 64, 107–123 (2005)
13. Dollar, P., Reabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: VS-PETS (2005)
14. Bissacco, A., Chiuso, A., Ma, Y., Soatto, S.: Recognition of human gaits. In: IEEE Conference on Computer Vision and Pattern Recognition (2001)
15. Bissacco, A., Chiuso, A., Soatto, S.: Classification and recognition of dynamical models: The role of phase, independent components, kernels and optimal transport. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(11), 1958–1972 (2007)
16. Basharat, A., Shah, M.: Time series prediction by chaotic modeling of nonlinear dynamical systems. In: International Conference on Computer Vision (2009)
17. Chaudhry, R., Ravichandran, A., Hager, G., Vidal, R.: Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
18. Kulis, B., Grauman, K.: Kernelized locality-sensitive hashing for scalable image search. In: International Conference on Computer Vision (2009)
19. Kulis, B., Darrel, T.: Learning to hash with binary reconstructive embeddings. Technical Report UCB/EECS-2009-101, Electrical Engineering and Computer Sciences, University of California at Berkeley (2009)
20. Subbarao, R., Meer, P.: Nonlinear mean shift over riemannian manifolds. International Journal of Computer Vision 84, 1–20 (2009)
21. Goh, A., Vidal, R.: Clustering and dimensionality reduction on Riemannian manifolds. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
22. Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning. Springer, Heidelberg (2003)
23. Schölkopf, B., Smola, A.: Learning with Kernels. MIT Press, Cambridge (2002)
24. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 888–905 (2000)
25. Ravichandran, A., Chaudhry, R., Vidal, R.: View-invariant dynamic texture recognition using a bag of dynamical systems. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
26. Çetingül, H.E., Vidal, R.: Intrinsic mean shift for clustering on stiefel and grassmann manifolds. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
27. Cock, K.D., Moor, B.D.: Subspace angles and distances between ARMA models. System and Control Letters 46, 265–270 (2002)
28. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local svm approach. In: International Conference on Pattern Recognition (2004)

The Quadratic-Chi Histogram Distance Family

Ofir Pele and Michael Werman

School of Computer Science
The Hebrew University of Jerusalem
{ofirpele,werman}@cs.huji.ac.il

Abstract. We present a new histogram distance family, the Quadratic-Chi (QC). QC members are Quadratic-Form distances with a cross-bin χ^2-like normalization. The cross-bin χ^2-like normalization reduces the effect of large bins having undo influence. Normalization was shown to be helpful in many cases, where the χ^2 histogram distance outperformed the L_2 norm. However, χ^2 is sensitive to quantization effects, such as caused by light changes, shape deformations etc. The Quadratic-Form part of QC members takes care of cross-bin relationships (*e.g.* red and orange), alleviating the quantization problem. We present two new cross-bin histogram distance properties: *Similarity-Matrix-Quantization-Invariance* and *Sparseness-Invariance* and show that QC distances have these properties. We also show that experimentally they boost performance. QC distances computation time complexity is linear in the number of non-zero entries in the bin-similarity matrix and histograms and it can easily be parallelized. We present results for image retrieval using the Scale Invariant Feature Transform (SIFT) and color image descriptors. In addition, we present results for shape classification using Shape Context (SC) and Inner Distance Shape Context (IDSC). We show that the new QC members outperform state of the art distances for these tasks, while having a short running time. The experimental results show that both the cross-bin property and the normalization are important.

1 Introduction

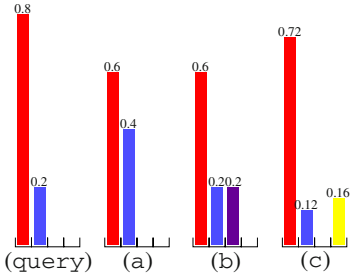
It is common practice to use bin-to-bin distances such as the L_1 and L_2 norms for comparing histograms. This practice assumes that the histogram domains are aligned. However this assumption is violated in many cases due to quantization, shape deformation, light changes, etc. Bin-to-bin distances depend on the number of bins. If it is low, the distance is robust, but not discriminative, if it is high, the distance is discriminative, but not robust. Distances that take into account cross-bin relationships (cross-bin distances) can be both robust and discriminative.

There are two kinds of cross-bin distances. The first is the Quadratic-Form distance [1]. Let P and Q be two histograms and A the bin-similarity matrix. The Quadratic-Form distance is defined as:

$$\mathrm{QF}^A(P,Q) = \sqrt{(P-Q)^T A (P-Q)} \tag{1}$$

When the bin-similarity matrix A is the inverse of the covariance matrix, the Quadratic-Form distance is called the Mahalanobis distance. If the bin-similarity matrix is positive-definitive, then the Quadratic-Form distance is a metric. In this case the

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 749–762, 2010.



distance	(a)-(c) ordered by their distance (in small fonts) to the (query)					
QCN(our)	(a)	0.35	(b)	0.62	(c)	0.86
QCS(our)	(a)	0.31	(b)	0.41	(c)	0.43
QF	(c)	0.20	(a)	0.28	(b)	0.28
EMD	(c)	3.20	(a)	4.00	(b)	4.00
L_1	(c)	0.32	(a)	0.40	(b)	0.40
χ^2	(a)	0.05	(c)	0.09	(b)	0.11

Fig. 1. This figure should be viewed in color, preferably on a computer screen. A toy example showing the behavior of distances that reduce the effect of large bins and the behavior of distances that take cross-bin relationships into account. We show four color histograms, each histogram has four colors: red, blue, purple, and yellow. The Quadratic-Form (QF), the Earth Mover Distance (EMD) and the L_1 norm do not reduce the effect of large bins. Thus, they rank (query) to be more similar to (c) than to (a). χ^2 considers (a) to be more similar, but as it does not take cross-bin relationships into account it fails with (b). Our proposed members of the Quadratic-Chi histogram distance family, QCN and QCS consider (a) to be most similar, (b) the second and (c) the least similar as they take into account cross-bin relationships and reduce the effect of large bins, using an appropriate normalization.

Quadratic-Form distance is the L_2 norm between linear transformations of P and Q. If the bin-similarity matrix is positive-semidefinite, then the Quadratic-Form distance is a semi-metric.

The second type of distance that takes into account cross-bin relationships is the Earth Mover's Distance (EMD). EMD was defined by Rubner et al. [2] as the minimal cost that must be paid to transform one histogram (P) into the other (Q):

$$\begin{aligned} \mathrm{EMD}^D(P,Q) &= (\min_{\{F_{ij}\}} \sum_{i,j} F_{ij} D_{ij}) / (\sum_{i,j} F_{ij}) \quad s.t \quad F_{ij} \geq 0 \\ \sum_j F_{ij} \leq P_i \quad \sum_i F_{ij} \leq Q_j \quad \sum_{i,j} F_{ij} &= \min(\sum_i P_i, \sum_j Q_j) \end{aligned} \tag{2}$$

where $\{F_{ij}\}$ denotes the flows. Each F_{ij} represents the amount transported from the ith supply to the jth demand. We call D_{ij} the *ground distance* between bin i and bin j. If D_{ij} is a metric, the EMD as defined by Rubner is a metric only for normalized histograms. Recently Pele and Werman [3] suggested $\widehat{\mathrm{EMD}}$:

$$\begin{aligned} \widehat{\mathrm{EMD}}^D_\alpha(P,Q) &= (\min_{\{F_{ij}\}} \sum_{i,j} F_{ij} D_{ij}) + |\sum_i P_i - \sum_j Q_j| \alpha \max_{i,j} D_{ij} \\ s.t \;\; &\text{EMD constraints} \end{aligned} \tag{3}$$

If D_{ij} is a metric and $\alpha \geq \frac{1}{2}$, $\widehat{\mathrm{EMD}}$ is a metric for all histograms [3]. For normalized histograms $\widehat{\mathrm{EMD}}$ and EMD are equal (*e.g.* Fig. 1).

In many natural histograms the difference between large bins is less important than the difference between small bins and should be reduced. See for example Fig. 1. The Chi-Squared (χ^2) is a histogram distance that takes this into account. It is defined as:

$$\chi^2(P,Q) = \frac{1}{2}\sum_i \frac{(P_i - Q_i)^2}{(P_i + Q_i)} \tag{4}$$

The χ^2 histogram distance comes from the χ^2 test-statistic [4] where it is used to test the fit between a distribution and observed frequencies. In this paper the histograms are not necessarily normalized, and thus not probabilities vectors. χ^2 was successfully used for texture and object categories classification [5,6,7], near duplicate image identification[8], local descriptors matching [9], shape classification [10,11] and boundary detection [12]. The χ^2, like other bin-to-bin distances such as the L_1 and the L_2 norms, is sensitive to quantization effects.

2 Our Contribution

In this paper we present a new cross-bin histogram distance family: Quadratic-Chi (QC). Like the Quadratic-Form, its members take cross-bin relationships into account. Like the χ^2, its members reduce the effect of differences caused by bins with large values. We discuss QC members' properties, including a formalization of a two new cross-bin histogram distance properties: *Similarity-Matrix-Quantization-Invariance* and *Sparseness-Invariance*. We show that all QC members and the EMD have these properties. We also show importance experimentally.

For full histograms QC distances computation time is linear in the number of non-zero entries in the bin-similarity matrix. In this case, QC distances can be implemented with 5 lines of Matlab code (see Algorithm 1). For two sparse histograms (for example bag-of-words histograms) with a total of S non-zeros entries and an average of K non-zeros entries in each row of the similarity matrix, a QC distance computation time complexity is $O(SK)$. See code (C++ and Matlab wrappers) at: http://www.cs.huji.ac.il/~ofirpele/QC/. Finally, QC distances' parallelization is trivial.

We present results for image retrieval on the Corel dataset using the SIFT descriptor [13] and small color images. We also present results for shape classification using

Algorithm 1. Quadratic-Chi Matlab Code for Full Histograms

```
function dist= QC(P,Q,A,m)

Z= (P+Q)*A;
% 1 can be any number as Z_i==0 iff D_i=0
Z(Z==0)= 1;
Z= Z.^m;
D= (P-Q)./Z;
% max is redundant if A is positive-semidefinite
dist= sqrt( max(D*A*D',0) );
```

Shape Context (SC) [10] and Inner Distance Shape Context (IDSC) [11]. QC members performance is excellent. They outperform state of the art distances including χ^2, QF, L_1, L_2, $\widehat{\text{EMD}}$[14], $\text{SIFT}_{\text{DIST}}$[3], EMD-$L_1$[15], Diffusion[16], Bhattacharyya [17], Kullback-Leibler[18] and Jensen-Shannon[19] while having a short running time. We have found that the normalization is very important. Surprisingly, excellent performance was achieved using a new bin-to-bin distance from the QC family, that has a large normalization factor. Its cross-bin version yielded an additional improvement, outperforming all other distances for SIFT, SC and IDSC.

3 The Quadratic-Chi Histogram Distance Family

3.1 The Quadratic-Chi Histogram Distance Definition

Let P and Q be two non-negative bounded histograms. That is, $P, Q \in [0, U]^N$. Let A be a non-negative symmetric bounded bin-similarity matrix such that each diagonal element is bigger or equal to every other element in its row (this demand is weaker than being a strongly dominant matrix). That is, $A \in [0, U]^N \times [0, U]^N$ and $\forall i, j\ A_{ii} \geq A_{ij}$. Let $0 \leq m < 1$ be the normalization factor. A Quadratic-Chi (QC) histogram distance is defined as:

$$\text{QC}_m^A(P,Q) = \sqrt{\sum_{ij} \left(\frac{(P_i - Q_i)}{\left(\sum_c (P_c + Q_c) A_{ci}\right)^m} \right) \left(\frac{(P_j - Q_j)}{\left(\sum_c (P_c + Q_c) A_{cj}\right)^m} \right) A_{ij}} \quad (5)$$

where we define $\frac{0}{0} = 0$. If A is positive-semidefinite, the argument inside the square root (the sum) is non-negative. If A is not positive-semidefinite we can get non-real (complex) distances. This is true also for the Quadratic-Form (Eq. 1). We prefer not to restrict ourselves to positive-semidefinite matrices. On the other hand, we don't want non-real distances. So, we define a complex distance as zero. In practice, this was never needed, even with non-positive-semidefinite matrices. This is due to the fact that the eigenvectors of the similarity matrices corresponding to negative eigenvalues were very far from smooth, while the difference vector for natural histograms P and Q is usually very smooth, see Fig. 2.

Each addend's denominator inside the square root is zero if and only if the addend's numerator is zero. A $\text{QC}_m^A(P,Q)$ distance is continuous. In particular, if the addend's denominator tends to zero, the whole addend tends to zero. Proofs are in [20].

The Quadratic-Chi distance family generalizes both the Quadratic-Form (QF) and a monotonic transformation of χ^2. That is, $\text{QC}_0^A(P,Q) = \text{QF}^A(P,Q)$ and if I is the identity matrix, $\text{QC}_{0.5}^I(P,Q) = \sqrt{2\chi^2(P,Q)}$.

3.2 Metric Properties

There are three conditions for a distance function, $\mathcal{D}$, to be a semi-metric. The first is *non-negativity* (*i.e.* $\mathcal{D}(P,Q) \geq 0$), the second is *symmetry* (*i.e.* $\mathcal{D}(P,Q) = \mathcal{D}(Q,P)$) and the third is *subadditivity* (*i.e.* $\mathcal{D}(P,Q) \leq \mathcal{D}(P,K) + \mathcal{D}(K,Q)$). $\mathcal{D}$ is a metric if it is

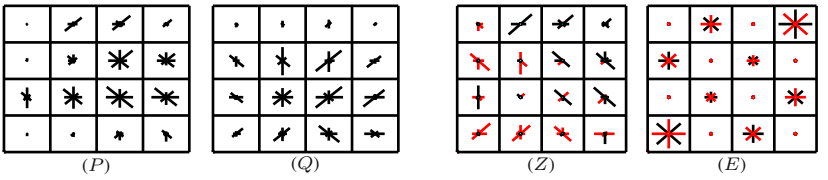


Fig. 2. This figure illustrates why it is not likely to get negative values in the square root argument of a QC distance for natural histograms and a typical similarity matrix. P and Q are two SIFT histograms. Z is the normalized difference vector. That is: $Z_i = \frac{P_i}{(\sum_c (P_c+Q_c)A_{ci})^m} - \frac{Q_i}{(\sum_c (P_c+Q_c)A_{ci})^m}$. Negative values are represented with red, positive values are represented with black. E is one of the eigenvectors of the similarity matrix that we used in the experiments which correspond to a negative eigenvalue. Z is very smooth while E is very non-smooth. This is typical of eigenvectors with negative values with typical parameters.

a semi-metric and it also has the property of *identity of indiscernibles* (*i.e.* $\mathcal{D}(P,Q) = 0$ if and only if $P = Q$).

A QC_m^A distance without the square root, is non-negative if the bin-similarity matrix, A, is positive-semidefinite. If A is positive-definitive, then it also has the property of *identity of indiscernibles*. This follows directly from the fact that the argument inside the square root in a QC histogram distance is a quadratic-form between two vectors. A QC histogram distance is symmetric if the bin-similarity matrix, A, is symmetric.

We now discuss subadditivity (*i.e.* $\mathcal{D}(P,Q) \leq \mathcal{D}(P,K) + \mathcal{D}(K,Q)$) for several distances. The χ^2 histogram distance is not subadditive. For example let $i = 0$, $k = 1$, $j = 2$ we get $\chi^2(i,j) = 1 > \chi^2(i,k) + \chi^2(k,j) = \frac{2}{3}$. However, $\sqrt{\chi^2}$ is subadditive for one and two dimensional non-negative histograms (verified by analysis). Experimentally it appears that $\sqrt{\chi^2}$ is subadditive for an N-dimensional non-negative histograms. Experimentally, QC members with the identity matrix seems to be subadditive for non-negative histograms. However, QC members with some positive-definitive bin-similarity matrices are not subadditive. The question when the QC histogram distances are subadditive is currently unresolved. An additional discussion about triangle inequality can be found in Jacobs et al. [21].

4 Cross-Bin Histogram Distance Properties

4.1 The Similarity-Matrix-Quantization-Invariance Property

The *Similarity-Matrix-Quantization-Invariance* property ensures that if two bins in the histograms have been erroneously quantized, this will not affect the distance. Mathematically we define this as:

Definition 1. *Let $\mathcal{D}$ be a cross-bin histogram distance between two histograms P and Q and let A be the bin-similarity/distance matrix. We assume P, Q and A are non-negative and that A is symmetric. Let $A_{k,:}$ be the kth row of A. Let $V = [V_1, \ldots, V_N]$*

be a non-negative vector and $0 \leq \alpha \leq 1$. *We define* $V^{\alpha,k,b} = [\ldots, \alpha V_k, \ldots, V_b + (1-\alpha)V_k, \ldots]$. *That is,* $V^{\alpha,k,b}$ *is a transformation of* V *where* $(1-\alpha)V_k$ *mass has moved from bin* k *to bin* b. *We define* $\mathcal{D}$ *to be* Similarity-Matrix-Quantization-Invariant *if:*

$$A_{k,:} = A_{b,:} \Rightarrow \forall\ 0 \leq \alpha \leq 1,\ 0 \leq \beta \leq 1\ \ \mathcal{D}^A(P,Q) = \mathcal{D}^A(P^{\alpha,k,b}, Q^{\beta,k,b}) \tag{6}$$

We prove that EMD, $\widehat{\text{EMD}}$ and all the Quadratic-Chi histogram distances are *Similarity-Matrix-Quantization-Invariant* in the appendix [20].

4.2 The Sparseness-Invariance Property

The *Sparseness-Invariance* property ensures that distances between sparse histograms will be equal to distances between full histograms. Mathematically we define this as:

Definition 2. *Let* $\mathcal{D}$ *be a cross-bin histogram distance between two histograms* $P \in \mathcal{R}^N$ *and* $Q \in \mathcal{R}^N$ *and let* A *be the* $N \times N$ *bin similarity/distance matrix. Let* A' *be any* $(N+1) \times (N+1)$ *matrix whose upper-left sub-matrix equals* A. *We define* $\mathcal{D}$ *to be* Sparseness-Invariant *if:*

$$\mathcal{D}^A([P_1, \ldots, P_n], [Q_1, \ldots, Q_n]) = \mathcal{D}^{A'}([P_1, \ldots, P_n, \mathbf{0}], [Q_1, \ldots, Q_n, \mathbf{0}]) \tag{7}$$

QC members, EMD and the $\widehat{\text{EMD}}$ are *Sparseness-Invariant* directly from their definitions. A stronger property called *Extension-Invariance* was proposed by D'Agostino and Dardanoni for bin-to-bin distances [22]. This property requires that, if both histograms are extended by concatenating each of them with the same vector (not necessarily zeros), the distance is left unaltered. Cross-bin distances assumes dependence between histogram bins, thus this requirement is too strong for them.

4.3 Cross-Bin Histogram Distance Properties Discussion

A *Sparseness-Invariant* cross-bin histogram distance does not depend on the specific representation of the histograms (full or sparse). A *Similarity-Matrix-Quantization-Invariant* cross-bin histogram distance encompass its cross-bin relationships only in the bin-similarity matrix. Intuitively such properties are desirable. In the appendix [20], we compare experimentally distances which resembles QC distances, but are either not *Similarity-Matrix-Quantization-Invariant* or not *Sparseness-Invariant*. The comparison shows that these properties considerably boost performance (especially for sparse color histograms).

Rubner et al. [2,23] claim that one of the key advantages of the Earth Mover's Distance is that each compared object may be represented by an individual (possibly with a different number of bins) binning that is adapted to its specific distribution. The Quadratic-Form is regarded as not having this property (see for example, Table 1 in [23]). Since all the Quadratic-Chi histogram distances (including the Quadratic-Form) are both *Similarity-Matrix-Quantization-Invariant* and *Sparseness-Invariant* there is no obstacle to using them with individual binning; *i.e.* to use them to compare histograms that were adapted to each object individually.

Similarity-Matrix-Quantization-Invariant and *Sparness-Invariant* can contradict. For example, any distance applied to the transformed vectors $P'_i = \sum_c (P_c) A_{ci}$ and $Q'_i = \sum_c (Q_c) A_{ci}$ is *Similarity-Matrix-Quantization-Invariant*. However the χ^2 distance between P' and Q' is not *Sparseness-Invariant* (with respect to P and Q).

5 Implementation Notes

5.1 The Similarity Matrix and the Normalization Factor

It is desirable to have a transformation from a distance matrix into a similarity matrix, as many spaces are equipped with a useful distance (*e.g.* color space [24]). Hafner et al. [1] proposed this transformation:

$$A_{ij} = 1 - \frac{D_{ij}}{\max_{ij}(D_{ij})} \tag{8}$$

Another possibility for choosing a similarity matrix is by using cross validation. However, we think that like for the Quadratic-Form, learning the similarity matrix (and for QC also the normalization factor) will be the best way to adjust them. This is left for future work. Currently we suggest to use thresholded ground distances as was used in [2,25,3,14] and choosing the normalization factor by cross validation.

5.2 Efficient Online Bin-Similarity Matrix Computation

For a fixed histogram configuration (*e.g.* SIFT, SC and IDSC) the bin-similarity matrix can be pre-computed once. Then, each distance computation is linear in the number of non-zero entries in the bin-similarity matrix.

There are cases where the bin-similarity matrix can not be pre-computed. For example, in our color experiments (Section 6.1), we used $N \times M$ color images as sparse histograms. That is, the query histogram was: $[1, \ldots, 1, 0, \ldots, 0]$ and each image being compared to the query was represented by the histogram: $[0, \ldots, 0, 1, \ldots, 1]$. Note that the full histogram dimension is $M \times N \times 256^3$, computing an $(M \times N \times 256^3)^2$ similarity matrix offline is not feasible. We can compute the similarity online for each pair of sparse histograms in $O((NM)^2)$ time. We now discuss how to do it more efficiently.

If we are comparing two images (as in Section 6.1) we can use a similarity matrix that gives far-away pixels zero similarity (see Eq. 10). Then, we can simply compare each pixel in one image to its corresponding $T \times T$ spatial neighbors in the second image. This reduces running time to $O(NMT^2)$. Using this technique, it is important to use a sparse representation for the bin-similarity matrix.

6 Results

We present results using the newly defined distances and state of the art distances, for image retrieval using SIFT-like descriptors and color image descriptors. In addition, we present results for shape classification using Inner Distance Shape Context (IDSC). More results for shape classification using SC, can be found in the appendix [20].

6.1 Image Retrieval Results

In this section we present results for image retrieval using the same benchmark as Pele and Werman [14]. We employed a database that contained 773 landscape images from the COREL database that were also used in Wang et al. [26]. The dataset has 10 classes[1]:

[1] The original database contains some visually ambiguous classes such as Africa that also contains images of beaches in Africa. We used the filtered image dataset that was downloaded from: http://www.cs.huji.ac.il/~ofirpele/FastEMD/

People in Africa, Beaches, Outdoor Buildings, Buses, Dinosaurs, Elephants, Flowers, Horses, Mountains and Food. The number of images in each class ranges from 50 to 100. From each class we selected 5 images as query images (images $1, 10, \ldots, 40$). Then we searched for the 50 nearest neighbors for each query image. We computed the distance of each image to the query image and its reflection and took the minimum. We present results for two types of image representations: SIFT-like descriptors and small L*a*b* images.

SIFT-like Descriptors. The first representation - SIFT is a $6 \times 8 \times 8$ SIFT descriptor [13] computed globally on the whole image. The second representation - CSIFT is a SIFT-like descriptor on a color-edge image. See [14] for more details.

We experimented with two new types of QC distances. The first is $\text{QC}^A_{0.5}$, which is a cross-bin generalization of $\sqrt{2\chi^2}$, which we call Quadratic-Chi-Squared (QCS). The second is $\text{QC}^A_{0.9}$, which has a larger normalization factor, which we call Quadratic-Chi-Normalized (QCN). We do not use QC^A_m with $m \geq 1$ due to discontinuity problems, see appendix [20] (practically, QC^A_1 had slightly poorer results compared to $\text{QC}^A_{0.9}$). We also experimented with the Quadratic-Form (QF) distance which is QC^A_0. For all of these distances we used the bin-similarity matrix in Eq. 8. Let $M = 8$ be the number of orientation bins, as in Pele and Werman [14], the ground distance between bins (x_i, y_i, o_i) and (x_j, y_j, o_j) is:

$$d_T(i,j) = \min\left(\left(||(x_i,y_i) - (x_j,y_j)||_2 + \min(|o_i - o_j|, M - |o_i - o_j|)\right) \ , \ T\right) \qquad (9)$$

We also used the identity matrix as a similarity matrix for all the above distances. We also compared to L_2 and χ^2. $\text{QF}^I = L_2$, and nearest neighbors of χ^2 and QCS^I are the same.

We also compared to four EMD variants. The first was $\widehat{\text{EMD}}^D_1$ with $D = d_T$ (Eq. 9) as in Pele and Werman [3]. The second was the L_1 norm which is equal to $\widehat{\text{EMD}}^D_{0.5}$ with D equals to the Kronecker delta multiplied by two. The third is $\text{SIFT}_{\text{DIST}}$ [3] which is the sum of $\widehat{\text{EMD}}$ over all the spatial cells (each spatial cell contains one orientation histogram). The ground distance for the orientation histograms is: $\min(|o_i - o_j|, M - |o_i - o_j|, 2)$ (M is the number of orientation bins). The fourth was the EMD-L_1 [15] which is EMD with L_1 as the ground distance. We also tried non-thresholded ground distances (which produce non-sparse similarity matrices). However, the results were poor. This is in line with Pele and Werman's findings that cross-bin distances should be used with thresholded ground distances [14]. Finally, we compared to the Diffusion distance proposed by Ling and Okada [16] and to three probabilistic based distances: Bhattacharyya [17], Kullback-Leibler (KL) [18] and Jensen-Shannon (JS) [19] (we added Matlab's epsilon to all histogram bins when computing KL and JS throughout the paper, as they are not well defined if there is a zero bin, without doing so accuracy was very low).

For each distance measure, we present the descriptor (SIFT/CSIFT) with which it performed best. The results for all the pairs of descriptors and distance measures can be found in the appendix[20]. The results are presented in Fig. 3(a) and show that $\text{QCN}^{1-\frac{d_{T=2}}{2}}$ (QCN with the similarity matrix: $A_{ij} = 1 - \frac{d_{T=2}(i,j)}{2}$) outperformed all other methods. $\widehat{\text{EMD}}^{d_{T=2}}_1$ ranked second. The computation of $\text{QCN}^{1-\frac{d_{T=2}}{2}}$ was 266

times faster than $\widehat{\text{EMD}}_1^{d_{T=2}}$, see Table 2 in page 760. QCN^I ranked third, which shows the importance of the normalization factor.

All cross-bin distances that use thresholded ground distances outperformed their bin-by-bin versions. The figure also shows that χ^2 and QF improve upon L_2. QCN and QCS which are mathematically sound combinations of χ^2 and QF outperformed both.

L*a*b* Images. Our second type of image representation is a small L*a*b* image. We resized each image to 32×48 and converted them to L*a*b* space. The state of the art color distance is Δ_{00} - CIEDE2000 on L*a*b* color space[24,27]. As it is meaningful only for small distances we threshold it (as in [2,25,14]).

Again, we experimented with QCS, QCN and QF distances using the bin-similarity matrix in Eq. 8. The ground distance between two pixels $(x_i, y_i, L_i, a_i, b_i)$, $(x_j, y_j, L_j, a_j, b_j)$:

$$\text{s}(i,j) = ||(x_i, y_i) - (x_j, y_j)||_2$$

$$\text{dc}_{T_1,T_2}(i,j) = \begin{cases} \min\left((\text{s}(i,j) + \Delta_{00}((L_i,a_i,b_i),(L_j,a_j,b_j))), T_1\right) & \text{if } \text{s}(i,j) \le T_2 \\ T_1 & \text{otherwise} \end{cases} \quad (10)$$

This distance is similar to the one used by [14], except that distances with spatial difference larger than the threshold T_2 are set to the maximum threshold T_1. This was done to accelerate the online computation of the bin-similarity matrix. The accuracy using this distance is the same as using the distance from Pele and Werman [14]. See appendix [20]. We also used $\widehat{\text{EMD}}$ with dc_{T_1,T_2} (Eq. 10) as a ground distance. Let I_1, I_2 be the two L*a*b* images. We also used the following distances:

$$L_1\Delta_{00} = \sum_{x,y}(\Delta_{00}(I_1(x,y), I_2(x,y))) \qquad L_1\Delta_{00}^T = \sum_{x,y}(\min(\Delta_{00}(I_1(x,y), I_2(x,y)), T))$$

$$L_2\Delta_{00} = \sum_{x,y}(\Delta_{00}(I_1(x,y), I_2(x,y)))^2 \qquad L_2\Delta_{00}^T = \sum_{x,y}(\min(\Delta_{00}(I_1(x,y), I_2(x,y)), T))^2$$

QCN^I, χ^2, L_2, L_1, $\text{SIFT}_{\text{DIST}}$[3], EMD-$L_1$[15], the Diffusion[16], Bhattacharyya [17], KL [18] and JS [19] distances cannot be applied to L*a*b* images as they are either bin-to-bin distances or applicable only to Manhattan networks.

We present results in Fig. 3. As shown, $\text{QCS}^{1-\frac{\text{dc}_{T_1=20,T_2=5}}{20}}$ and $\widehat{\text{EMD}}_1^{\text{dc}_{T_1=20,T_2=5}}$ [14] distances ranked first. $\text{QCS}^{1-\frac{\text{dc}_{T_1=20,T_2=5}}{20}}$ ran 300 times faster (see Table 2). However, since the computation of the bin-similarity matrix cannot be offline here, the real gain is a factor of 17. The $\text{QF}^{1-\frac{\text{dc}_{T_1=20,T_2=5}}{20}}$ distance ranked last, which shows the importance of the normalization factor of the QC histogram members.

Although a QC distance alleviates quantization problems, EMD does it better, instead of matching everything to everything it finds the optimal matching. EMD however, does not reduce the effect of large bins. We conjecture that a variant of EMD which will reduce the effect of large bins will have an excellent performance.

6.2 Shape Classification Results

In this section we present results for shape classification using the same framework as Ling et al. [11,15,28]. We test for shape classification with the Inner Distance Shape

Table 1. Shape classification results. $\text{QCN}^{1-\frac{\text{dsc}_T=2}{2}}$ outperformed all other distances.

	Top 1	Top 2	Top 3	Top 4	AUC%
$\text{QCN}^{1-\frac{\text{dsc}_T=2}{2}}$	39	38	38	34	0.950
QCN^{I}	40	37	36	33	0.940
$\text{QCS}^{1-\frac{\text{dsc}_T=2}{2}}$	39	35	38	28	0.912
QCS^{I}	40	34	37	27	0.907
χ^2	40	36	36	21	0.902
$QF^{1-\frac{\text{dsc}_T=2}{2}}$	40	34	39	19	0.897
L_2	39	35	35	18	0.873

	Top 1	Top 2	Top 3	Top 4	AUC%
$\widehat{\text{EMD}}_1^{\text{dsc}T=2}$	39	36	35	27	0.902
L_1	39	35	35	25	0.890
$\text{SIFT}_{\text{DIST}}$ [3]	38	37	27	22	0.848
EMD-L_1 [15]	39	35	38	30	0.917
Diffusion [16]	39	35	34	23	0.880
Bhattacharyya [17]	40	37	32	23	0.895
KL [18]	40	38	36	29	0.938
JS [19]	40	35	37	21	0.900

Context (IDSC) [11]. The original Shape Context (SC) descriptor was proposed by Belongie et al. [10]. Belongie et al. [10] and Ling and Jacobs [11] used the χ^2 distance for comparing shape context histograms. Ling and Okada [15] showed that replacing χ^2 with EMD-L_1 improves results. We show that QC members yields the best results.

We tested on the articulated shape data set [11,28], that contains 40 images from 8 different objects. Each object has 5 images articulated to different degrees. The dataset is very challenging because of the similarity between different objects. The original SC had a very poor performance on this dataset, see appendix [20].

Again, we experimented with QCS, QCN and QF distances with the bin-similarity matrix in Eq. 8. The ground distance between two bins (r_i, o_i), (r_i, o_i) was (M is the number of orientation bins):

$$\text{dsc}_T(i,j) = \min\left((|d_i - d_j| + \min(|o_i - o_j|, M - |o_i - o_j|), T\right) \tag{11}$$

We also used the identity matrix as a similarity matrix, and thus we also compare to L_2. χ^2 and QCS^I distances are not equivalent here as the distance is not used for nearest neighbors. We refer the reader to Belongie et al. paper to see its usage [10]. Practically, QCS^I slightly outperformed χ^2 in this task, see Table 1.

We also compared to four EMD variants: $\widehat{\text{EMD}}_1^D$ with $D = \text{dsc}_T$ (Eq. 11), the L_1 norm, $\text{SIFT}_{\text{DIST}}$ [3] and EMD-L_1 [15]. Finally, we compared to the Diffusion distance proposed by Ling and Okada [16] and to three probabilistic based distances: Bhattacharyya [17], Kullback-Leibler (KL) [18] and Jensen-Shannon (JS) [19].

To evaluate results, for each image, the four most similar matches are chosen from other images in the dataset. The retrieval result is summarized as the number of 1st, 2nd, 3rd and 4th most similar matches that come from the correct object. Table 1 shows the retrieval results. The $\text{QCN}^{1-\frac{\text{dsc}_T=2}{2}}$ outperformed all the other methods. QCN^I performance is again excellent, which shows the importance of the normalization factor.

Again all cross-bin distances outperformed their bin-by-bin versions. Again, χ^2 and QF improved upon L_2. QCN and QCS which are mathematically sound combinations of χ^2 and QF outperformed both.

6.3 Running Time Results

All runs were conducted on a Pentium 2.8GHz. A comparison of the practical running time of all distances is given in Table 2. Clearly QCN and QCS distances are fast to

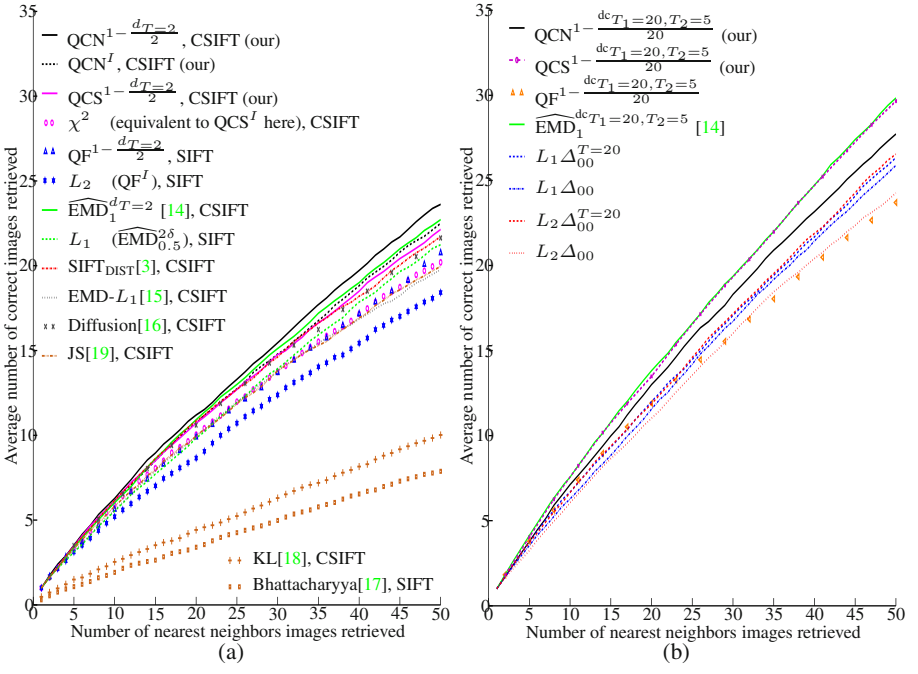


Fig. 3. Results for image retrieval.
(a) **SIFT-like descriptors.** For each distance measure, we present the descriptor (SIFT/CSIFT) with which it performed best. The results for all the pairs of descriptors and distance measures can be found in the appendix[20]. There are several key observations. First, the QC members performance is excellent. $\mathrm{QCN}^{1-\frac{d_{T=2}}{2}}$ (QCN with the similarity matrix: $A_{ij} = 1 - \frac{d_{T=2}(i,j)}{2}$) outperformed all other distances. $\widehat{\mathrm{EMD}}_1^{d_{T=2}}$ ranked second, but its computation was 266 times slower than $\mathrm{QCN}^{1-\frac{d_{T=2}}{2}}$ computation (see Table 2). Second, all cross-bin versions of the distances (with d_T or a transformation of it) performed better than their bin-by-bin versions (with the identity matrix or the Kronecker delta function). Third, QCN^I ranked third, although its a bin-to-bin distance. This shows the importance of the normalization factor. Finally, χ^2 and QF improve upon L_2. However, χ^2 does not take cross-bin relationships into account and QF does not reduce the effect of large bins. QCS and QCN histogram distances, which are mathematically sound combinations of χ^2 and QF have the two properties and outperformed both.
(b) **L*a*b* images results.** QCN^I, χ^2, L_2, L_1, $\mathrm{SIFT}_{\mathrm{DIST}}$[3], EMD-$L_1$[15], Diffusion[16], Bhattacharyya[17], KL[18] and JS[19] distances are not applicable here. $\mathrm{QCS}^{1-\frac{\mathrm{dc}_{T_1=20,T_2=5}}{20}}$ and $\widehat{\mathrm{EMD}}_1^{\mathrm{dc}_{T_1=20,T_2=5}}$ [14] ranked first. $\mathrm{QCS}^{1-\frac{\mathrm{dc}_{T_1=20,T_2=5}}{20}}$ computation is 300 times faster than $\widehat{\mathrm{EMD}}_1^{\mathrm{dc}_{T_1=20,T_2=5}}$ without taking the bin-similarity matrix computation into account and 17 times faster when it is taken into account (see Table 2). $\mathrm{QF}^{1-\frac{\mathrm{dc}_{T_1=20,T_2=5}}{20}}$ ranked last, which shows the importance of the normalization factor in QC members.

Table 2. (SIFT) 384-dimensional SIFT-like descriptors matching time (in *milliseconds*). The distances from left to right are the same as the distances in Fig. 3 (a) from up to down.
(IDSC) 60-dimensional IDSC histograms matching time (in *microseconds*). The distances from left to right are the same as the distances in Table 1 from up to down.
(L*a*b*) 32×48 L*a*b* images matching time (in *milliseconds*). The distances from left to right are the same as the distances in Fig. 3 (b) from up to down. In parentheses is the time it takes to compute the distance and the bin-similarity matrix as it cannot be computed offline.

Descriptor	QCN^{A_2}	QCN^{I}	QCS^{A_2}	QCS^{I}	χ^2	QF^{A_2}	L_2	$\widehat{\text{EMD}}^{D_2}$ [14]	L_1	$\text{SIFT}_{\text{DIST}}$ [3]
(SIFT)	**0.15**	**0.1**	**0.07**	**0.014**	**0.013**	**0.05**	**0.011**	**40**	**0.011**	**0.07**
(IDSC)	**6.41**	**2.99**	**2.32**	**0.35**	**0.34**	**1.25**	**0.14**	**133.75**	**0.32**	**0.31**

Descriptor	EMD-L_1 [15]	Diffusion [16]	JS [19]	KL [18]	Bhattacharyya [17]
(SIFT)	**40**	**0.27**	**0.088**	**0.048**	**0.015**
(IDSC)	**20.57**	**3.15**	**1.40**	**8.53**	**17.17**

Descriptor	$\text{QCN}^{A_{20}}$	$\text{QCS}^{A_{20}}$	$\text{QF}^{A_{20}}$	$\widehat{\text{EMD}}^{D_{20}}$ [14]	$L_1\Delta_{00}^{T=20}$	$L_1\Delta_{00}$	$L_2\Delta_{00}^{T=20}$	$L_2\Delta_{00}$
(L*a*b*)	**20** (370)	**19** (369)	**11** (361)	**6000** (6350)	**3.2**	**3.2**	**3.2**	**3.2**

compute. This is consistent with their linear time complexity. The only non-linear time distances are $\widehat{\text{EMD}}$ [14] and EMD-L_1 [15] which are also practically much slower than the other methods. Our method can be easily parallelized, taking advantage of multi-core computers or the GPU.

7 Conclusions

We presented a new cross-bin distance family - the Quadratic-Chi (QC). QC distances have many desirable properties. Like the Quadratic-Form histogram distance they take into account cross-bin relationships. Like χ^2 they reduce the effect of large bins. We formalized two new cross-bin properties, *Similarity-Matrix-Quantization-Invariance* and *Sparseness -Invariance*. QC members were shown to have both. Finally, QC distance computation time is linear in the number of non-zero entries in the bin-similarity matrix. Experimentally, QC outperformed state of the art distances, while having a very short run-time.

There are several open questions that we still need to explore. The first is for which QC distances does the the triangle inequality holds for. The second is whether we can change the Earth Mover's Distance so that it will also reduce the effect of large bins. Concave-cost network flow [29] seems to be the right direction for future work although it presents two major obstacles. First, the concave-cost network flow optimization is NP-hard [29]. However, there are available approximations [29,30]. Second, simply using concave-cost flow networks will result in a distance which is not *Similarity-Matrix-Quantization-Invariant*. We would also like to explore whether metric learning methods such as [31,32,33,34,35,36,37,38] can be generalized for the Quadratic-Chi histogram distance. Assent et al. [39] have suggested methods that accelerate database retrieval

that uses Quadratic-Form distances. Generalizing these methods for the Quadratic-Chi distances is of interest. Finally, other computer vision applications such as tracking can use the QC distances. The project homepage, including code (C++ and Matlab wrappers) is at: http://www.cs.huji.ac.il/~ofirpele/QC/.

References

1. Hafner, J., Sawhney, H., Equitz, W., Flickner, M., Niblack, W.: Efficient color histogram indexing for quadratic form distance functions. PAMI (1995)
2. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. IJCV (2000)
3. Pele, O., Werman, M.: A linear time histogram metric for improved sift matching. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 495–508. Springer, Heidelberg (2008)
4. Snedecor, G., Cochran, W.: Statistical Methods, Ames, Iowa, 6th edn. (1967)
5. Cula, O., Dana, K.: 3D texture recognition using bidirectional feature histograms. IJCV (2004)
6. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. IJCV (2007)
7. Varma, M., Zisserman, A.: A statistical approach to material classification using image patch exemplars. PAMI (2009)
8. Xu, D., Cham, T., Yan, S., Duan, L., Chang, S.: Near Duplicate Identification with Spatially Aligned Pyramid Matching. In: CSVT (accepted)
9. Forssén, P., Lowe, D.: Shape Descriptors for Maximally Stable Extremal Regions. In: ICCV (2007)
10. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. PAMI (2002)
11. Ling, H., Jacobs, D.: Shape classification using the inner-distance. PAMI (2007)
12. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. PAMI (2004)
13. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV (2004)
14. Pele, O., Werman, M.: Fast and robust earth mover's distances. In: ICCV (2009)
15. Ling, H., Okada, K.: An Efficient Earth Mover's Distance Algorithm for Robust Histogram Comparison. PAMI (2007)
16. Ling, H., Okada, K.: Diffusion distance for histogram comparison. In: CVPR (2006)
17. Bhattacharyya, A.: On a measure of divergence between two statistical populations defined by their probability distributions. BCMS (1943)
18. Kullback, S., Leibler, R.: On information and sufficiency. AMS (1951)
19. Lin, J.: Divergence measures based on the Shannon entropy. IT (1991)
20. Pele, O., Werman, M.: The quadratic-chi histogram distance family - appendices (2010), http://www.cs.huji.ac.il/ ofirpele/publications/ECCV2010app.pdf
21. Jacobs, D., Weinshall, D., Gdalyahu, Y.: Classification with nonmetric distances: Image retrieval and class representation. PAMI (2000)
22. D'Agostino, M., Dardanoni, V.: What's so special about Euclidean distance? SCW (2009)
23. Rubner, Y., Puzicha, J., Tomasi, C., Buhmann, J.: Empirical evaluation of dissimilarity measures for color and texture. CVIU (2001)
24. Luo, M., Cui, G., Rigg, B.: The Development of the CIE 2000 Colour-Difference Formula: CIEDE2000. CRA (2001)

25. Ruzon, M., Tomasi, C.: Edge, Junction, and Corner Detection Using Color Distributions. PAMI (2001)
26. Wang, J., Li, J., Wiederhold, G.: SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture LIbraries. PAMI (2001)
27. Sharma, G., Wu, W., Dalal, E.: The CIEDE2000 color-difference formula: implementation notes, supplementary test data, and mathematical observations. CRA (2005)
28. Ling, H.: Articulated shape benchmark and idsc code (2010), http://www.ist.temple.edu/ hbling/code/inner-dist-articu-distrbution.zip
29. Guisewite, G., Pardalos, P.: Minimum concave-cost network flow problems: Applications, complexity, and algorithms. AOR (1990)
30. Amiri, A., Pirkul, H.: New formulation and relaxation to solve a concave-cost network flow problem. JORS (1997)
31. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning with application to clustering with side-information. In: NIPS (2003)
32. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning distance functions using equivalence relations. In: ICML (2003)
33. Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R.: Neighbourhood components analysis. In: NIPS (2005)
34. Globerson, A., Roweis, S.: Metric learning by collapsing classes. In: NIPS (2006)
35. Yang, L., Jin, R.: Distance metric learning: A comprehensive survey. MSU (2006)
36. Davis, J., Kulis, B., Jain, P., Sra, S., Dhillon, I.: Information-theoretic metric learning. In: ICML (2007)
37. Yu, J., Amores, J., Sebe, N., Radeva, P., Tian, Q.: Distance learning for similarity estimation. PAMI (2008)
38. Weinberger, K., Saul, L.: Distance metric learning for large margin nearest neighbor classification. JMLR (2009)
39. Assent, I., Wichterich, M., Seidl, T.: Adaptable Distance Functions for Similarity-based Multimedia Retrieval. DSN (2006)

Membrane Nonrigid Image Registration

Geoffrey Oxholm and Ko Nishino

Department of Computer Science
Drexel University
Philadelphia, PA

Abstract. We introduce a novel nonrigid 2D image registration method that establishes dense and accurate correspondences across images without the need of any manual intervention. Our key insight is to model the image as a membrane, i.e., a thin 3D surface, and to constrain its deformation based on its geometric properties. To do so, we derive a novel Bayesian formulation. We impose priors on the moving membrane which act to preserve its shape as it deforms to meet the target. We derive these as curvature weighted first and second order derivatives that correspond to the changes in stretching and bending potential energies of the membrane and estimate the registration as the maximum a posteriori. Experimental results on real data demonstrate the effectiveness of our method, in particular, its robustness to local minima and its ability to establish accurate correspondences across the entire image. The results clearly show that our method overcomes the shortcomings of previous intensity-based and feature-based approaches with conventional uniform smoothing or diffeomorphic constraints that suffer from large errors in textureless regions and in areas in-between specified features.

1 Introduction

The goal of nonrigid image registration is to align a template image to a reference image by locally deforming the template image. Modeling nonlinear, local deformations has important applications in many computer vision problems including image stabilization [1], subject tracking [2,3], and medical imaging [4], to name a few.

There are two primary approaches to nonrigid image registration: intensity-based and feature-based. Intensity-based approaches [5,6,7] attempt to minimize the intensity differences across the entire image. Such methods produce dense correspondences but suffer from ambiguities arising from similar intensity regions. Feature-based methods [8,9,10] compute deformations that align a sparse set of specifically selected features. These points are then used in conjunction with a parametric model to interpolate the recovered deformations across the rest of the image. In addition to the separate challenge of detecting and matching good features (which often relies on manual intervention), the overall quality of the registration directly relies on the interpolation method. Consequently, accuracy inherently decays rapidly as the distance from the feature points increases.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 763–776, 2010.

In this paper, we introduce an automatic nonrigid 2D image registration method that establishes dense and accurate correspondences across the entire image without the need to provide feature correspondences a priori. Our key idea is to model the image as a 2D membrane embedded in a 3D spatial-intensity space. We then formulate nonrigid image registration as the process of aligning two membranes by deforming one to the other while preserving its local geometric structures. In particular, we model the elastic and bending potential energies of the membrane. By penalizing their changes, the local structures of the template membrane are preserved as it deforms to meet the reference membrane.

We derive a probabilistic formulation of this membrane nonrigid image registration. We model each template image point as a Gaussian and seek the maximum a posteriori estimate of the template image as a mixture of Gaussians given the reference image. Our main contributions are a newly derived likelihood and priors that reflect physically-motivated constraints on the membrane geometry:

Novel likelihood: We construct a Gaussian at each pixel of the template image scaled by the membrane's original curvature at that point. This naturally encodes the significance of the underlying image structure, which in turn encourages features to align with corresponding features.

Bending energy: We model the inherent flexibility of a membrane by penalizing local surface deformations in proportion to the membrane's original curvature. This corresponds to minimizing the change in potential bending energy which translates into a novel curvature-weighted second order derivative prior.

Stretching energy: We model the inherent elasticity of a membrane by penalizing surface stretching and compression. This corresponds to minimizing the change in potential elastic energy across the membrane which translates into a novel first order derivative prior.

Intuitively, this formulation leads to surface regions with prominent local structures (features of the membrane) to be preserved and aligned with each other while more smooth regions are allowed to deform more flexibly. By preserving the shape of the membrane features, their appearance in the image being modeled remain true to their underlying geometry.

We demonstrate the accuracy and effectiveness of our method on 2D slices of real brain MRIs and images of faces with different expressions. In particular, we show that in addition to a significant decrease in overall intensity error, our method establishes accurate correspondences of prominent image structures automatically. This has strong implications in various applications since local image structures usually correspond to meaningful geometric structures of the imaged scene or object, and accurately aligning such structures is of great importance.

2 Related Work

Nonrigid image registration has been a popular area of research. Here we focus on methods that specifically address the shortcomings of both intensity-based and feature-based methods. We refer the reader to surveys of the rich literature [11,12,4] for more thorough context.

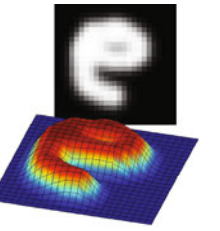
Fischer and Modersitzki [13] combine the two approaches on a sliding scale. They initially register a set of manually established features, then incrementally shift towards a uniform intensity-based metric. Our curvature-scaled objective function has a similar effect in that it encourages the rapid registration of feature rich areas. It does so, however, without requiring predefined features or by giving priority to the registration of any subregion.

Fischer and Modersitzki [14] also introduced a "curvature-based" normalization term that encourages locally smooth deformations by penalizing sharp changes in the displacement field. Although we also describe our bending energy constraint term as "curvature-based," the two approaches are fundamentally different. Whereas their normalization term is a second-order derivative of the 2D displacement field, we impose an energy minimization prior on the membrane, i.e., the image modeled as a 3D spatial-intensity surface. This added dimension allows us to impose geometrically-induced constraints on the image deformation.

Intensity-based methods assume that corresponding regions in the imaged scene maintain the same intensity pattern in both images. Previous authors [15,16] have noted that this assumption can lead to violations of the basic physical properties of the subject which are present despite changes in illumination. To address this they use mass or volumetric constraints specific to their given applications. More general methods like Thirion's Demons method [5] and the recent diffeomorphic extension of this work by Vercauteren et al. [7] smooth the 2D deformation field thereby preventing large feature displacements from tearing or folding the deformation field. Although smooth deformation fields are found, ambiguities arising from similar intensity patterns of non-corresponding regions result in undesirable non-local artifacts. In our physically motivated model, we avoid such local minima by preserving the shape of the image membrane thereby maintaining local structures as they move across the image. To address folding we introduce a novel prior which allows pixels to come quite close to each other without overlapping. This allows us to model the common physical occurrence of creasing which is impossible under the various smoothing models.

Recently, probabilistic formulations of nonrigid image registration have gained further attention. Jian and Vemuri [17] use a Gaussian mixture model to register two point sets by placing a Gaussian at each point. Our work is most closely related to the extension of this approach by Myronenko et al. [6] that formulates image registration as a Gaussian mixture estimation with Gaussians centered on each pixel. By placing quadratic priors on each Gaussian, they preserve the distance of each pixel to its neighbors thereby avoiding tearing and folding of the deformation field. This results in a locally smooth deformation with well-minimized intensity distance on synthetic deformations. Unavoidably, however, these priors perform less well on real-world data which exhibit more complex transformations that cannot be modeled with assumptions of smoothness. Since accurate correspondences are of primary concern in many applications, we show that the minimization of an intensity distance is an insufficient objective function without shape preserving constraints.

3 Bayesian Membrane Registration



We model the image as a 2D membrane in a 3D space. In order to ensure this membrane approximates the actual imaged surface, the intensities are normalized and the height of each pixel is set proportionately to the log of the normalized intensity.[1] As noted by Koenderink and van Doorn [18], by using the log-intensity we ensure a geometrically invariant intensity encoding which eliminates any effect intensity magnitude may otherwise have on our geometric constraints while simultaneously achieving a degree of symmetry between the Cartesian pixel coordinates and heights of the points on the membrane.

More precisely, we view the image coordinates $\widehat{\mathbf{x}} = (\widehat{\mathbf{x}}_u, \widehat{\mathbf{x}}_v)$ and scaled logarithm of the normalized intensity of each point $I(\widehat{\mathbf{x}})$ together as points $\mathbf{x} = (\widehat{\mathbf{x}}_u, \widehat{\mathbf{x}}_v, I(\widehat{\mathbf{x}}))$ of a 2D membrane in a 3D space. In many cases we may assume that this membrane reflects the geometry of the imaged object. For instance, a Lambertian surface would have its normals roughly encoded in its shading and the intensity in medical images reflects the density of the subject.

Similar to Myronenko et al. [6], we formulate nonrigid image registration as a MAP estimation of a product of Gaussian mixture densities. The posterior, representing the probability of the template image $\mathbf{Y}$ given the reference image $\mathbf{X}$ and parameters θ, is formulated as

$$p(\mathbf{Y}|\mathbf{X}, \theta) \propto p(\mathbf{X}|\mathbf{Y}, \theta) p(\mathbf{Y}|\theta) \ , \tag{1}$$

where we assume uniform normalization $p(\mathbf{X})$. We have five parameters, $\theta = (\mathbf{Y}^0, \sigma_0, \beta_\mathrm{e}, \beta_\mathrm{b}, \beta_\mathrm{f})$, which we describe below. Here $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^\mathrm{T}$ is an $N \times 3$ matrix containing the points in the reference membrane $\mathbf{x} = (\widehat{\mathbf{x}}_u, \widehat{\mathbf{x}}_v, I(\widehat{\mathbf{x}}))$ and $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_M)^\mathrm{T}$ is an $M \times 3$ matrix containing the final locations of the registered template membrane's points $\mathbf{y} = (\widehat{\mathbf{y}}_u, \widehat{\mathbf{y}}_v, I(\widehat{\mathbf{y}}))$. We denote the original, undeformed template membrane as $\mathbf{Y}^0 = (\mathbf{y}^0_1, \ldots, \mathbf{y}^0_M)^\mathrm{T}$. N and M are the number of pixels in the images (which need not be equal in size). We model the likelihood as a product of N independent Gaussian mixture densities $p(\mathbf{X}|\mathbf{Y}, \theta) = \prod_n p(\mathbf{x}_n|\mathbf{Y})$. Building on this formulation, we introduce a curvature-based scaling to each point as we discuss next.

Our key contributions lie in the three priors on the template membrane $\mathbf{Y}$,

$$p(\mathbf{Y}|\theta) \propto \exp\left(-\beta_\mathrm{e}\mathcal{E}(\mathbf{Y}) - \beta_\mathrm{b}\mathcal{B}(\mathbf{Y}) - \beta_\mathrm{f}\mathcal{F}(\mathbf{Y})\right) \ . \tag{2}$$

The first, $\mathcal{E}(\cdot)$, quantifies the amount of change in elastic potential energy in the membrane. The second, $\mathcal{B}(\cdot)$, quantifies the change in bending potential energy in the membrane. Finally, $\mathcal{F}(\cdot)$ quantifies the amount of folding, or overlap, in the membrane. Each function is weighted by a parameter $\beta_{\{\mathrm{e,b,f}\}} \in \theta$. We will now describe each component of the posterior in more detail.

[1] Results are consistent so long as the images are normalized consistently. Since this formulation is geometrically invariant, the scale only effects the convergence rate.

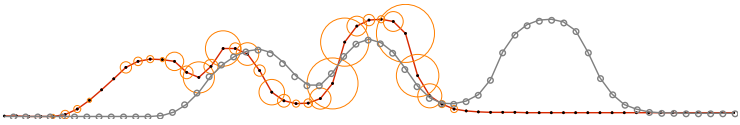


Fig. 1. Shown in this 1D example, a Gaussian is established for each pixel of the template image (solid) with standard deviation (circles) proportional to the curvature at that point. This allows prominent local structures that usually have high curvature to travel further and align with corresponding structures of the reference image (dotted) while preserving their shapes as modeled by their elastic and bending energies.

3.1 Gaussian Mixture Likelihood

The mixture density $p(\mathbf{x}_n)$ for a pixel of the reference image $\mathbf{x}_n$ is expressed probabilistically as a Gaussian mixture where each point of the template membrane is expressed as its own Gaussian distribution

$$p(\mathbf{x}_n|\mathbf{Y}) = \sum_{m=1}^{M} \frac{1}{M} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \ . \tag{3}$$

Observing that regions with prominent local structures (features) are more indicative of the membrane's overall shape, we allow points in these regions a larger range of motion by scaling the Gaussian centered at each point by the membrane's original curvature at that point. Using the squared mean curvature $H^2(\mathbf{y}_m^0)$ we model this with a per-point mean and standard deviation of

$$\boldsymbol{\mu}_m = \mathbf{y}_m, \qquad \boldsymbol{\Sigma}_m = (H^2(\mathbf{y}_m^0)\sigma_0)^2\mathbf{I}_3 \ , \tag{4}$$

where $\mathbf{I}_3$ is the identity matrix as each image dimension is statistically independent. These feature rich areas maintain their shape due to increased rigidity constraints (discussed in the next section). Intuitively, this leads to feature-rich surface regions to be preserved and aligned with each other, guiding the registration of the rest of the membrane.

We can then express the likelihood across the entire image as an unweighted product of these Gaussian mixture densities

$$p(\mathbf{X}|\mathbf{Y}, \sigma_0) \propto \prod_{n=1}^{N} \sum_{m=1}^{M} \exp\left[-\frac{1}{2} \left\| \frac{\mathbf{x}_n - \mathbf{y}_m}{H^2(\mathbf{y}_m^0)\sigma_0} \right\|^2 \right] \ . \tag{5}$$

In other words, for a given scale parameter σ_0, the final pixel locations $\mathbf{Y}$ that maximize this likelihood represent the deformation that maps the points of the template membrane to regions of the reference membrane.

In Fig. 1 we show a simple one-dimensional, $(x, I(x))$, example where the initial template $\mathbf{Y}^0$ is shown in red, and the reference $\mathbf{X}$ is shown in gray. The relative standard deviations of the Gaussians are shown as orange circles. As shown in Fig. 2, this increase in the search space for key regions of the curve is necessary to avoid local minima and preserve the geometry of membrane features.

Fig. 2. Two results after registering the curves of Fig. 1 are shown (solid) relative to the target curve (dotted). Standard smoothing priors [6] (*left*) can cause local minima to be found. Here an entire peak is unregistered while two peaks have collapsed into one. Imposing our physically-based constraints (*right*) ensures that the structure of the entire curve is maintained during deformation resulting in a more accurate registration.

The shape of the deformed membrane must now be considered. Without constraints on the deformation, the pixel locations can be permuted at will to maximize the likelihood. To ensure an accurate deformation, we introduce physically-motivated priors that operate on the local geometry of the membrane.

3.2 Shape-Preserving Priors

The membrane model of an image allows us to incorporate physically-based constraints that preserve the local intensity structures of the image as it deforms. In particular, we model the elastic and bending potential energies of the membrane and impose geometric constraints that minimize the changes in these energies.

Elastic Energy. The elastic energy of a deformation captures the change in elastic potential as the membrane deforms. We define this energy as the sum of the elastic energy across all points $\mathcal{E}(\mathbf{Y}) = \sum_m \mathcal{E}(\mathbf{y}_m)$. We define the elastic energy at a point as the change in elastic potential energy at that point $\mathbf{y}$ relative to the potential at that point in the original membrane $\mathbf{y}^0$. We evaluate the potential of a point on a membrane using Hooke's law $E = \frac{1}{2}kx^2$. By assuming the elastic constant (k in Hooke's law) is uniform across the membrane we let $\beta_\mathrm{e} = (k/2)^2$ which is then used to weight the entire energy term. The relative displacement (x in Hooke's law) at each point naturally corresponds to the total change in distance to the point's neighbors $\mathrm{ne}(\mathbf{y})$.

By squaring the difference in potential of the relaxed and deformed membranes, we naturally quantify the amount of elastic energy at each point as

$$\mathcal{E}(\mathbf{y}) = \sum_{\mathbf{y}_i \in \mathrm{ne}(\mathbf{y})} (\|\mathbf{y}_i - \mathbf{y}\|^2 - \|\mathbf{y}_i^0 - \mathbf{y}^0\|^2)^2. \tag{6}$$

Note that because the intensity of a pixel does not change this reduces to

$$\mathcal{E}(\mathbf{y}) = \sum_{\mathbf{y}_i \in \mathrm{ne}(\mathbf{y})} (\|\widehat{\mathbf{y}}_i - \widehat{\mathbf{y}}\|^2 - 1)^2. \tag{7}$$

This prior differs considerably from stretching or elastic constraints of past work. Specifically, the first-order smoothing terms used in past work impose smoothing on the 2D deformation field itself, necessarily resulting in overly smooth local deformations.

Bending Energy. We also model the bending potential energy of the membrane and derive an energy term which quantifies the change in this potential as the membrane deforms. We define the total bending energy as the sum across all points, $\mathcal{B}(\mathbf{Y}) = \sum_m \mathcal{B}(\mathbf{y}_m)$. Our bending potential function is based on the Willmore energy $\int_S \frac{1}{2}H^2 - KdA$, where H is the mean curvature function and K is the Gaussian curvature function. By the Gauss-Bonnet theorem K is a topological invariant, and so remains constant during the deformation. Since we are concerned with the change in this energy, this term cancels out. We extend the Willmore energy to include the inherent rigidity of structural features by considering the potential of each point separately.

Whereas homogeneous membranes have uniform elasticity, the flexibility of a membrane varies with the curvature of the undeformed surface [19]. This translates to weighting the bending energy with a per-point rigidity coefficient equal to the squared mean curvature of the undeformed membrane at that point $H^2(\mathbf{Y}^0)$. This term also provides robustness to noise since a corrupt pixel will yield a high curvature value at that point. Since mean curvature is computationally expensive, we use the Laplacian $\Delta(\cdot)$ as an approximation for $H^2(\cdot)$ when computing the change in energy [20]. We define the bending energy as the weighted squared change in bending potential

$$\mathcal{B}(\mathbf{y}) = H^2(\mathbf{y}^0)\left(\Delta(\mathbf{y}) - \Delta(\mathbf{y}^0)\right)^2 . \tag{8}$$

At a given point, the Laplacian of a surface is expressed using the (log) intensity heights $I(\cdot)$ of the point $\mathbf{y} = (\widehat{\mathbf{y}}_u, \widehat{\mathbf{y}}_v, I(\widehat{\mathbf{y}}))$ and its negative direction and positive direction neighbors $\mathbf{y}_-$ and $\mathbf{y}_+$ respectively

$$\Delta(\mathbf{y}) = \left(\frac{h_-(I(\widehat{\mathbf{y}}_+) - I(\widehat{\mathbf{y}})) - h_+(I(\widehat{\mathbf{y}}) - I(\widehat{\mathbf{y}}_-))}{h_+ h_- h_\pm}\right)^2 , \tag{9}$$

where the distance to the positive direction neighbor h_+ the negative direction neighbor h_- and the distance between midpoints $h_\pm$ are used

$$h_+ = \|\widehat{\mathbf{y}}_+ - \widehat{\mathbf{y}}\| \ , \quad h_- = \|\widehat{\mathbf{y}} - \widehat{\mathbf{y}}_-\| \ , \quad h_\pm = \|[(\widehat{\mathbf{y}}_+ + \widehat{\mathbf{y}}) - (\widehat{\mathbf{y}} + \widehat{\mathbf{y}}_-)]/2\| \ . \tag{10}$$

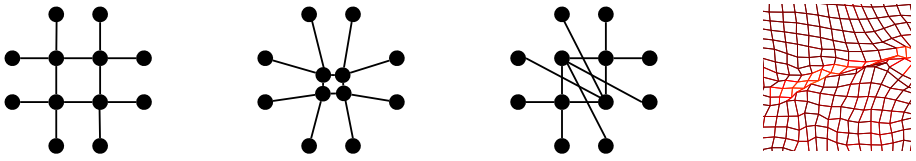
As a time saving approximation we assume $h_+ = h_- = h_\pm$. We also note that the numerator is equal for $\Delta(\mathbf{y})$ and $\Delta(\mathbf{y}^0)$ since the intensities of the pixels do not change. Further, we note that $h_\pm^0$ is constant which allows us to reduce the horizontal bending penalization of Eq. 8 to

$$\mathcal{B}(\mathbf{y}) \propto H^2(\mathbf{y}^0)\left(h_\pm^{-2} - 1\right)^2 . \tag{11}$$

For 2D images we consider horizontal, vertical, and two diagonal bending energies by formulating $\mathcal{B}_\rightarrow$, $\mathcal{B}_\downarrow$, $\mathcal{B}_\nearrow$, and $\mathcal{B}_\searrow$ analogously and take the sum

$$\mathcal{B}(\mathbf{y}) = \mathcal{B}_\rightarrow(\mathbf{y}) + \mathcal{B}_\downarrow(\mathbf{y}) + \mathcal{B}_\nearrow(\mathbf{y}) + \mathcal{B}_\searrow(\mathbf{y}) \ . \tag{12}$$

Folding Prior. During registration, regions of the deforming template membrane will expand and compress to meet the corresponding reference regions. As



(a) $\mathcal{E} = 0.0$ (b) $\mathcal{E} = 7.7$ (c) $\mathcal{E} = 5.7$ (d) Crease

Fig. 3. The elastic penalization for areas of compression is minimized when neighboring surface patches fold over one another in featureless regions. We address this with an explicit prior on folding which allows for creases to form but eliminates folding.

shown in Fig. 3, since our elastic energy constraint encourages uniform spacing and our bending energy constraint applies primarily to feature rich areas, folding can occur. Although the bending prior discourages this in textured areas, it is not sufficient in relatively featureless regions.

Conventional methods decrease this by imposing second order derivative penalizations on the 2D deformation field [6,14] or by specifically modeling diffeomorphic registrations [7]. Problems arise, however, in regions that change in size dramatically. As real-world objects inevitably experience such large deformations, a more accurate model should allow sharp boundaries in the deformation field as neighboring regions converge and creases form.

We allow such sharp boundaries to form with an explicit model of folding that allows pixels to come quite close to each other without penalty while strongly penalizing folding. We model this with a sigmoid function on each of the four neighboring directions of a point $\mathbf{y} = (\widehat{\mathbf{y}}_u, \widehat{\mathbf{y}}_v, I(\widehat{\mathbf{y}}))$. The folding energy of a deformation is then the sum across the deformation of each of these four values

$$\mathcal{F}(\mathbf{Y}) = \sum_{m=1}^{M} \left(\mathcal{F}_{\rightarrow}(\mathbf{y}_m) + \mathcal{F}_{\leftarrow}(\mathbf{y}_m) + \mathcal{F}_{\uparrow}(\mathbf{y}_m) + \mathcal{F}_{\downarrow}(\mathbf{y}_m) \right) . \tag{13}$$

For example, the right neighbor function is given by

$$\mathcal{F}_{\rightarrow}(\mathbf{y}) = \left(1 + \exp\{c(\widehat{\mathbf{y}}_u^{+} - \widehat{\mathbf{y}}_u + t)\}\right)^{-1}, \tag{14}$$

where $\widehat{\mathbf{y}}^{+}$ is the right neighbor of $\widehat{\mathbf{y}}$. We establish the other three functions similarly. In this formulation a sufficiently high value for c and low value for t effectively make this a step function that penalizes the folding of neighboring pixels while allowing pixels to form sharp boundaries without penalty.

3.3 MAP Estimation

Having formulated the likelihood and prior constraints, we may estimate the maximum a posteriori using energy minimization. Specifically, the log posterior

$$\log p(\mathbf{Y}|\mathbf{X},\theta) = \sum_{n=1}^{N} \log \sum_{m=1}^{M} e^{-\frac{1}{2}\left\| \frac{\mathbf{x}_n - \mathbf{y}_m}{H(\mathbf{y}_m^0)\sigma_0} \right\|^2} - \beta_{\mathrm{e}}\mathcal{E}(\mathbf{Y}) - \beta_{\mathrm{b}}\mathcal{B}(\mathbf{Y}) - \beta_{\mathrm{f}}\mathcal{F}(\mathbf{Y}) + C \tag{15}$$

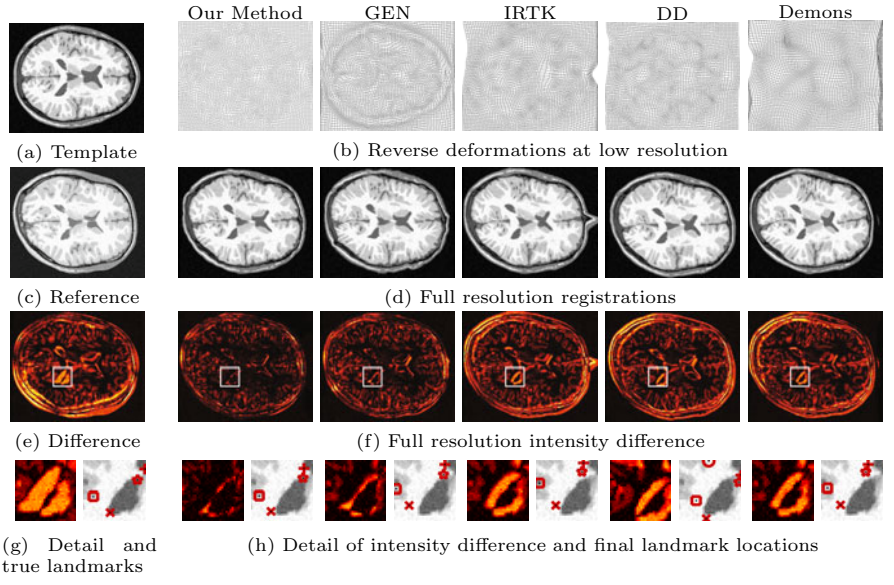


Fig. 4. Template and reference images (from BrainWeb [22]) are scaled down and registration is performed with various methods. The resulting reverse deformation grid (b) is applied to the original template image (a). These registrations (d) are subtracted from the reference image (c). The error is then visualized (f) and compared with the difference of the original template and reference images (e). Increased brightness corresponds to larger error. Detailed inspections (h) of a region requiring a large transformation (g) show that our method results in the least error both in terms of the intensity difference and in the alignment accuracy of features. This example is labeled "Brain1" in Fig. 8.

can be maximized using simulated annealing over the scale parameter σ_0 [6]. We vary σ_0 between σ_{max} and σ_{min} which depend only on the size of the images and are set automatically. The solution for each iteration is found with an interior trust region method [21]. In practice our rigidity constraints have proven robust to large values for σ_{max}. Typically $\sigma_{min} = 0.5$, $\sigma_{max} = 6$, and 6 annealing iterations are needed to converge for 100×100 images. We set $t = 1$ and $c = 5$ in our folding prior $\mathcal{F}$ to allow faster convergence of each annealing iteration. With this smooth penalization, however, resulting registrations occasionally have some amount of folding of the registration. To address this, after each annealing iteration points that have folded over each other are merged together. Unfortunately, each iteration is still quite computationally expensive, requiring as much as forty-five minutes in our current unoptimized implementation. We envision a significant speed-up with an approximate linearization of the objective function.

4 Experimental Results

We evaluate the accuracy of our model on human facial expressions and 2D slices of real brain MRIs. We compare the results with four characteristic automatic

methods. Rueckert et al. [23] introduced an intensity-based approach that uses b-splines to smooth the deformation field which they released as part of their Image Registration Toolkit (IRTK). Thiron's well known Demons method [5], uses gradient information from the reference image to determine the amount of force the deforming points must exert. This work was later extended by Vercauteren et al. [7] to specifically model diffeomorphisims in a model termed Diffeomorphic Demons (DD). Finally, we compare our work to the generalized elastic net (GEN) model of Myronenko et al. [6] that uses a Gaussian mixture formulation similar to ours but with conventional smoothing priors. When possible, we use publicly available implementations of these algorithms with default parameters.

Past work use synthetic deformations to compare their results to ground-truth deformations. Synthetic deformations, however, are generated without regard for the physical structure of the image subject and therefore provide little information about real-world accuracy. Instead, we observe that an ideal registration should conform to the structural properties of the imaged subject. A deformation field embodying this characteristic should therefore maintain accuracy even when applied to a higher resolution image. At this increased resolution we may then compare the intensity error as well as the locations of manually labeled feature points to test sub-pixel accuracy. What may be termed "under-fitting" or "over-fitting" occurs when a deformation field appears well-suited at one resolution, but reveals significant inaccuracy at higher resolutions.

In Fig. 4 we compare the results of our method on a subject from the BrainWeb database [22] with the other methods. A lower resolution version of the template image (4a) is registered to an equally down-sampled reference image (4c). The resulting inverse deformation fields (4b) show where each pixel in the resulting registration originated. The resulting high resolution registrations (4d), formed using a bilinearly interpolated inverse deformation field, are then compared with the reference image (4c) and the absolute difference is visualized as a heat map (4f) in which the brightness of the pixel increases as the error increases. Our approach produces a significantly improved registration, as evident by the greater amount of black (4f). Closer inspection (4h) shows the feature alignment accuracy of our method as evident by the close proximity of the feature points to key anatomical landmarks (4g). Here we also see that the error for this region is less than the interpolation scale (which is 3 in this case), revealing the degree of sub-pixel accuracy of our method. This example is labeled "Brain1" in Fig. 8. Note that we achieve a minimum 29% decrease in overall intensity error, while achieving a 29% decrease in feature alignment error.

Fig. 5 details the importance of shape preservation. When using a smoothing model, a landmark that was originally at the tip of a long feature looses this distinction and becomes embedded in a mass. With our method, the local geometry of the feature is preserved and a more accurate registration is achieved.

In Fig. 6 we qualitatively compare the registrations a neutral face (6a) to a smiling face (6b) from the Japanese Female Facial Expressions (JAFFE) database [24]. In analyzing the deformation fields we find that the key challenge in expression registration is the sudden appearance of dark regions that were not

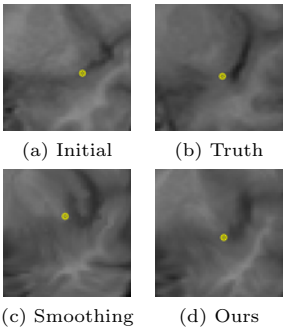


Fig. 5. Our shape preserving priors (d) ensure that meaningful correspondences are made as compared to smoothing models (5) (c)

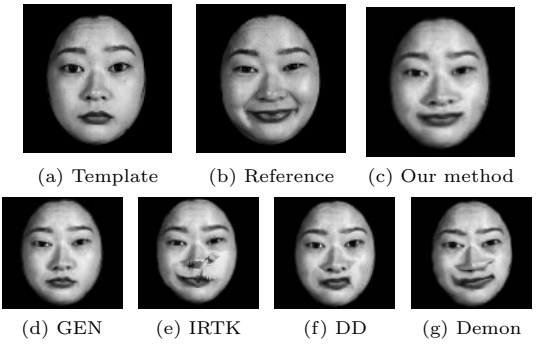


Fig. 6. Faces (from JAFFE [24]) present a particular challenge due to dramatic local deformations and intensity variations in corresponding regions like the creases of a smile. Our method outperforms past work by preserving the shape of the features as they deform. This example is "Face1" in the graphs of Fig. 8.

previously present. In this example the formation of a smile introduces dramatic changes in brightness in the cheeks as creases appear. This causes gross deformations to result in the other, less-structured models. Our method, on the other hand, achieves a much more accurate registration across the entire face. Although it is not possible to recreate the creases without changing the intensity of the pixels, the shape of the lips and raise in cheeks are captured well.

Three more datasets are shown in Fig. 7. The first example shows the registration of a neutral face to a frown. Note how as the upper lip compresses, a crease forms in the chin. The co-appearance of these dark regions above and below the original lip creates a challenging ambiguity. IRTK tries to split this dark region between the lip and chin whereas the Diffeomorphic Demons method shifts the mouth down to meet the chin crease. Our method achieves an accurate result by maintaining the membrane geometry of the whole mouth as it stretches and curves down. The second example shows the registration of a neutral face to a sad face. This subtle expression demonstrates the key criticism of intensity-based methods – despite the reasonable appearance of these results each method fails to accurately align key landmarks as evidenced by the bright sections of error surrounding the facial features exhibited by every method except ours. Note also how the lower lip has been dramatically compressed in various methods to meet the highlight that shows up in the reference image. The final example shows two horizontal MRI slices of the same subject. Note how the top left portion of the ventricle has become almost completely occluded in the reference image. As is shown in the detail of Fig. 3d, our method correctly models this as a crease whereas the other models have no way to register a feature that has disappeared.

In Fig. 8 we quantitatively evaluate our results and compare them with these methods. For each dataset cluster the yellow column (labeled "None") shows the value of the error measures when no algorithm is used indicating the relative

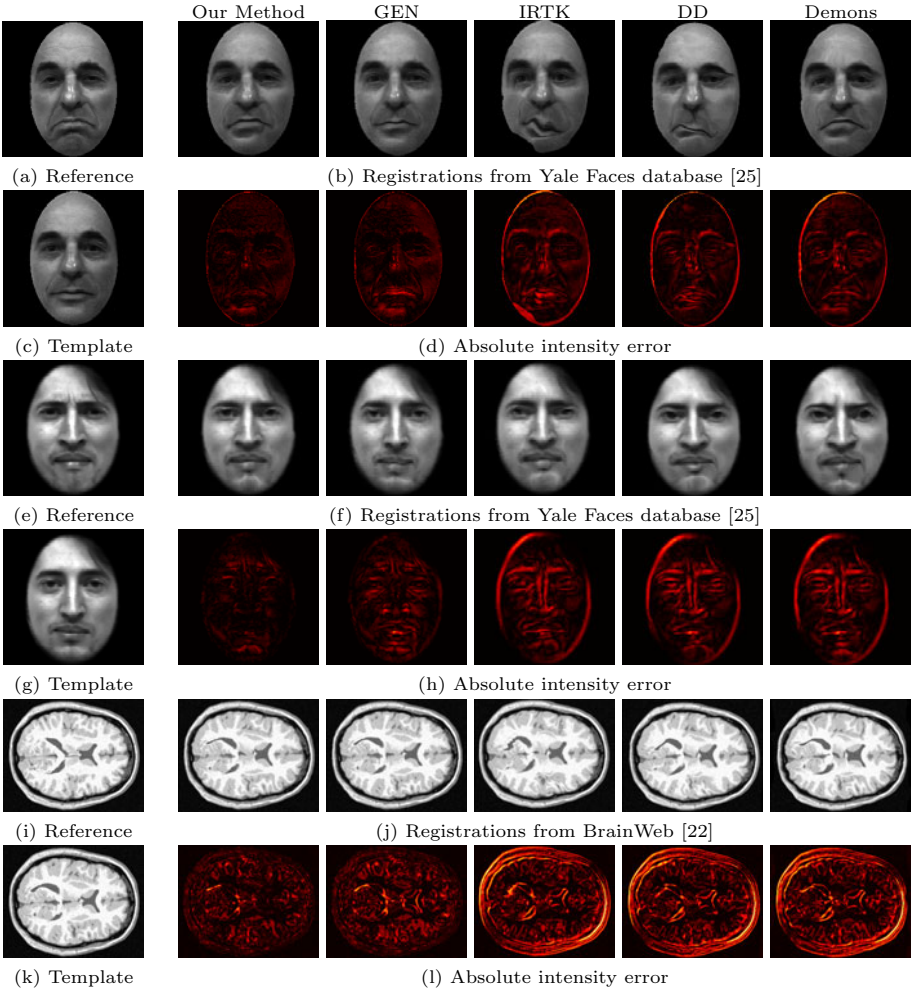
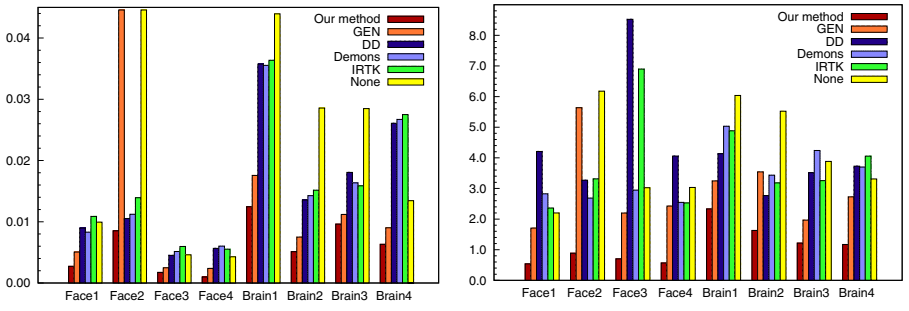


Fig. 7. Facial expressions and 2D MRI slices are registered using our method, GEN [6], IRTK [23], Diffeomorphic Demons (DD) [7], and Demons [5] methods respectively. These datasets are labeled Face3, Face4, and Brain4, in the graphs of Fig. 8

magnitude of the measures. In Fig. 8a we compare the mean squared error in intensity $[0, 1]$ of the registration. The results show that our method results in an average of 32% less error than the next best method. In Fig 8b we compare the mean error of final landmark locations (in pixels) from the manually labeled ground-truth locations. For each image pair we annotate between 12 and 27 primary feature correspondences. Here our method achieves an average of 53% less error in feature alignment than the next best method with values consistently below the interpolation scale.



(a) Mean squared intensity error

(b) Mean landmark error (in pixels)

Fig. 8. In both graphs the yellow (right-most) bar of each grouping indicates the amount of error if no registration is performed. Our method consistently outperforms every other benchmark method.

5 Conclusion

Our method demonstrates considerable accuracy that results from our key assumption – that the image as a membrane in 3D spatial-intensity space approximates the actual surface of the subject and preserving its geometric shape reflects the true image deformation more accurately. Experimental results have shown that in many cases the assumption is valid and geometrically induced constraints increase accuracy dramatically. In particular, our method achieves higher accuracy in both the overall deformation and resulting feature correspondences. The resulting registrations exhibit a robustness to the common pitfalls of intensity-based registration techniques while maintaining particularly high accuracy for feature points automatically. This has strong implications in various applications where the accuracy of correspondences is particularly important.

Acknowledgements. This work was supported in part by National Science Foundation CAREER Award IIS-0746717 and IIS-0803670.

References

1. Irani, M., Rousso, B., Peleg, S.: Recovery of Ego-Motion Using Image Stabilization. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 454–460 (1994)
2. Dedeoglu, G., Kanade, T., Baker, S.: The Asymmetry of Image Registration and its Application to Face Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 807–823 (2007)
3. Weese, J., Penney, G., Desmedt, P., Buzug, T.M., Hill, D., Hawkes, D.: Voxel-Based 2-D/3-D Registration of Fluoroscopy Images and CT Scans for Image-Guided Surgery. IEEE Trans. on Info. Technology in Biomedicine 1, 284–293 (1997)
4. Maintz, J., Viergever, M.: A Survey of Medical Image Registration. ACM Computing Surveys 2, 1–36 (1998)
5. Thirion, J.P.: Non-rigid Matching Using Demons. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 245–251 (1996)

6. Myronenko, A., Song, X., Carreira-Perpinán, M.A.: Free-Form Nonrigid Image Registration Using Generalized Elastic Nets. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
7. Vercauteren, T., Pennec, X., Perchant, A., Ayache, N.: Diffeomorphic Demons: Efficient Non-parametric Image Registration. NeuroImage 45, S61–S72 (2009)
8. Wang, K., He, Y., Qin, H.: Incorporating Rigid Structures in Non-rigid Registration Using Triangular B-Splines. Variational, Geometric, and Level Set Methods in Computer Vision 3752, 235 (2005)
9. Kohlrausch, J., Rohr, K., Stiehl, H.: A New Class of Elastic Body Splines for Nonrigid Registration of Medical Images. Journal of Mathematical Imaging and Vision 23, 280 (2005)
10. Davis, M.H., Khotanzad, A., Flamig, D.P., Harms, S.E.: Elastic Body Splines: A Physics Based Approach to Coordinate Transformation in Medical Image Matching. In: IEEE Symposium on Computer-Based Medical System, vol. 8, p. 81 (1995)
11. Zitova, B., Flusser, J.: Image Registration Methods: A Survey. Image and Vision Computing 21, 977–1000 (2003)
12. Brown, L.G.: A Survey of Image Registration Techniques. ACM Computing Surveys 24, 325–376 (1992)
13. Fischer, B., Modersitzki, J.: Combination of Automatic Non-rigid and Landmark Based Registration: The Best of Both Worlds. Society of Photo-Optical Instrumentation Engineers 5032, 1037–1048 (2003)
14. Fischer, B., Modersitzki, J.: Curvature Based Image Registration. Journal of Mathematical Imaging and Vision 18, 81–85 (2003)
15. Haker, S., Tannenbaum, A., Kikinis, R.: Mass Preserving Mappings and Image Registration. In: Niessen, W.J., Viergever, M.A. (eds.) MICCAI 2001. LNCS, vol. 2208, pp. 120–127. Springer, Heidelberg (2001)
16. Haber, E., Modersitzki, J.: Volume preserving image registration. In: Barillot, C., Haynor, D.R., Hellier, P. (eds.) MICCAI 2004. LNCS, vol. 3216, pp. 591–598. Springer, Heidelberg (2004)
17. Jian, B., Vemuri, B.C.: A Robust Algorithm for Point Set Registration Using Mixture of Gaussians. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, p. 1246 (2005)
18. Koenderink, J., van Doorn, A.: Image Processing Done Right. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 158–172. Springer, Heidelberg (2002)
19. Grinspun, E., Hirani, A., Desbrun, M., Schröder, P.: Discrete Shells. In: ACM Special Interest Group on Graphics and Interactive Techniques, p. 67 (2003)
20. Wardetzky, M., Bergou, M., Harmon, D., Zorin, D., Grinspun, E.: Discrete Quadratic Curvature Energies. Comp. Aided Geom. Design 24, 499–518 (2007)
21. Coleman, T.F., Li, Y.: An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds. SIAM Journal on Optimization 6, 418–445 (1996)
22. Cocosco, C., Kollokian, V., Kwan, R., Pike, G.B., Evans, A.C.: Brainweb: Online Interface to a 3D MRI Simulated Brain Database. Functional Mapping of the Human Brain 5, 425 (1997)
23. Rueckert, D., Sonoda, L., Hayes, C., Hill, D., Leach, M., Hawkes, D.: Nonrigid Registration Using Free-Form Deformations: Application to Breast MR Images. IEEE Transactions on Medical Imaging 18, 712–721 (1999)
24. Lyons, M., Akamatsu, S., Kamachi, M., Gyoba, J.: Coding Facial Expressions With Gabor Wavelets. In: Face and Gesture Recognition, p. 200 (1998)
25. Yale: Face Database (1997), http://cvc.yale.edu/projects/yalefaces/yalefaces.html

Affine Puzzle: Realigning Deformed Object Fragments without Correspondences*

Csaba Domokos and Zoltan Kato

Department of Image Processing and Computer Graphics,
University of Szeged
H-6701 Szeged, PO. Box 652., Hungary
Fax: +36 62 546-397
{dcs,kato}@inf.u-szeged.hu

Abstract. This paper is addressing the problem of realigning broken objects without correspondences. We consider linear transformations between the object fragments and present the method through 2D and 3D affine transformations. The basic idea is to construct and solve a polynomial system of equations which provides the unknown parameters of the alignment. We have quantitatively evaluated the proposed algorithm on a large synthetic dataset containing 2D and 3D images. The results show that the method performs well and robust against segmentation errors. We also present experiments on 2D real images as well as on volumetric medical images applied to surgical planning.

1 Introduction

In this paper we address the problem of reassembling an object from its parts. This is also known as the *puzzle* problem, which is not only interesting from a theoretical point of view [1,2], but also arises in many application domains like archaeology [3] or medical imaging [4] *e.g.* bone fracture reduction [5,6,7]. The affine puzzle problem can be formulated as follows: Given a binary image of an object (the *template*) and another binary image (the *observation*) containing the fragments of the *template*, we want to establish the geometric correspondence between these images which reconstructs the complete *template* object from its parts. The overall distortion is a global nonlinear transformation with the following constraint: 1) the object parts are distinct (*i.e.* either disconnected or separated by segmentation), 2) all fragments of the *template* are available, but 3) each of them is subject to a *different* affine deformation.

A related problem is partial matching of shapes [8,9,10]. Partial matching addresses a particularly challenging setting of classical shape matching, where

* This research was partially supported by the Hungarian Scientific Research Fund (OTKA) – K75637 and by the grant CNK80370 of the National Office for Research and Technology (NKTH) & the Hungarian Scientific Research Fund (OTKA). The CT images were obtained from the University of Szeged, Department of Trauma Surgery and were used with permission of Prof. Endre Varga, MD.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 777–790, 2010.

two shapes are dissimilar in general, but have significant similar parts. In this context, our problem would require to find a partial matching between the *template* and each fragments of the *observation*. Current approaches are usually based on the Laplace-Beltrami framework [11,10], but classical approaches like the Iterative Closest Point (ICP) [12] algorithm can also be used assuming an appropriate shape representation [8]. Considering the rather high computational complexity of these algorithms, this solution is far from optimal for our problem.

Another related problem is the piece-wise approximation of nonlinear deformations by locally linear transformations. In [13], the distortion is modeled as locally affine but globally smooth transformation, which accounts for local and global variations in image intensities. The classical solution [14] comprises identifying point correspondences based on salient points between the images and then either a time consuming optimization procedure or the solution of a system of equations provide the parameters of the unknown deformation. Finding reliable point correspondences between the images is a difficult problem on its own.

Most of the existing solutions to the puzzle problem [1,2,3] consist in matching fragment-pairs to find neighbors, which are then reassembled by a rigid body transformation. In [1], Kong and Kimia propose a 2D curve matching technique based on the geometric features of puzzle pieces. The solution is obtained by a recursive grouping of triples using a best-first search strategy. The method can be extended to 3D fragments scanned by a laser range finder, where a pair of ridges are matched using a generalization of the 2D curve matching approach. In [3], the rather high computational complexity of curve matching is reduced by adopting a multiscale technique. Papaioannou *et al.* address the problem of 3D object reconstruction using only the surface geometry of fragments, assuming no information about the final model to be reconstructed [2]. The basic idea of the method is that the best fit of two 3D fragments is likely to occur at their relative pose, which minimizes the point-by-point distance between the mutually visible faces of the fragments. Matched pieces are then glued via a rigid-body transformation.

Although classical approaches may account for a *template* object by incorporating a set of constraints to improve the overall performance, they are primarily targeted to problems where a *template* is not available, *e.g.* archaeology [3]. On the other hand, there are many applications where a *template* object is available: In industrial applications usually 3D models of manufactured parts can be easily produced. In medical imaging an *atlas* can be used or, by taking advantage of the symmetry of the human body, the intact bone can provide a *template* for bone fracture reduction, as shown in Section 5.3. Therefore we address this important setting of the puzzle problem and propose a generic solution which is then applied to 2D and 3D transformations. The methodology adopted here is similar in spirit to the affine matching methods of [15] and [16]. However, none of these works addresses the puzzle problem. [16] assumes that both images contain the same number of shapes and radiometric information is available. Based on these informations, Hagege and Francos construct a linear system of equations which provides the parameters of the aligning transformations. Since the

partitioning of the *template* is not available, this method cannot be used here. In [15], Domokos and Kato presented an elegant solution to recover affine deformations between 2D shapes. This method is also unable to solve the puzzle problem because the deformation is nonlinear and there is no direct correspondence between the *template* and its observed fragments.

In Section 2, a general solution is proposed followed by Section 3 about the numerical implementation issues and Section 4 presenting the application of our method for various linear transformations. Finally, Section 5 presents quantitative results on 2D and 3D synthetic datasets as well as on various real images and Section 6 concludes the paper.

2 Realigning Object Parts

Given an n dimensional *template* object and an *observation* containing its affine deformed fragments, we want to recover the transformations realigning these shapes into their original position on the *template*. Let us denote the homogeneous point coordinates of the *template* and *observation* by $\mathbf{x} = [x_1, \ldots, x_n, 1]$ and $\mathbf{y} = [y_1, \ldots, y_n, 1] \in \mathbb{P}^n$. Furthermore, let $\ell \in \mathbb{N}$ denote the number of fragments on the *observation*. The transformation aligning the *observation* with the *template* is a non-linear one, composed of ℓ linear transformations

$$\mathbf{A}_i = \begin{bmatrix} a_{i11} & a_{i12} & \ldots & a_{i1(n+1)} \\ \vdots & & \ddots & \vdots \\ a_{in1} & a_{in2} & \ldots & a_{in(n+1)} \\ 0 & 0 & \ldots & 1 \end{bmatrix} \quad i = 1, \ldots \ell\,. \tag{1}$$

Since the *observation* has disjoint parts, we can assume that points of each deformed shape are labeled by the function $\lambda' : \mathbb{P}^n \to \{0, 1, \ldots, \ell\}$, which assigns 0 to the background. Obviously, there is a corresponding *hidden* labeling $\lambda : \mathbb{P}^n \to \{0, 1, \ldots, \ell\}$ which assigns the label i to the *template* points corresponding to the i^{th} shape. Our goal is to recover the affine matrices $\{\mathbf{A}_i\}_{i=1}^{\ell}$. The main challenges are that neither the partitioning (*i.e.* the hidden labeling λ) of the *template* nor correspondences between the shapes are known.

2.1 Solution for One Pair of Shapes

Let us first establish a solution for the i^{th} shape. The *template* and *observation* domains are denoted by $\mathcal{D}_i = \{\mathbf{x} \in \mathbb{P}^n | \lambda(\mathbf{x}) = i\}$ and $\mathcal{D}'_i = \{\mathbf{y} \in \mathbb{P}^n | \lambda'(\mathbf{y}) = i\}$, respectively. Note that $\mathcal{D}'_i$ is known but $\mathcal{D}_i$ is unknown. The points of these domains are related by the unknown transformation $\mathbf{A}_i$:

$$\mathbf{x} = \mathbf{A}_i \mathbf{y}. \tag{2}$$

One way to recover $\mathbf{A}_i$ is to establish point correspondences and then set up a system of equations from Eq. (2). Since $\mathcal{D}_i$ is unknown, finding correspondences is practically impossible. Therefore we are interested in a direct method without

solving the correspondence problem. For that purpose, let us notice that that the relation in Eq. (2) remains valid when a function $\omega : \mathbb{P}^n \to \mathbb{R}$ is acting on both sides of the equation [15]:

$$\omega(\mathbf{x}) = \omega(\mathbf{A}_i\mathbf{y}) \ . \tag{3}$$

We then integrate out individual point correspondences [15] yielding

$$\int_{\mathcal{D}_i} \omega(\mathbf{x})d\mathbf{x} = |\mathbf{A}_i| \int_{\mathcal{D}'_i} \omega(\mathbf{A}_i\mathbf{y})d\mathbf{y}, \tag{4}$$

where the integral transformation $\mathbf{x} = \mathbf{A}_i\mathbf{y}$, $d\mathbf{x} = |\mathbf{A}_i|d\mathbf{y}$ has been applied, and $|\mathbf{A}_i|$ is the Jacobian determinant. Based on Eq. (4), we can construct as many equations as needed by making use of a set of linearly independent functions $\{\omega_j\}_{j=1}^m$, $m \geq n(n+1)$. The solution of the resulting nonlinear system of equations provides the parameters of $\mathbf{A}_i$ [15].

2.2 Solving for All Shapes Simultaneously

We have established relations between the i^{th} shape-pair, but we know neither the correspondence between the shapes nor the partitioning $\mathcal{D}_i$ of the *template*. Would these information available, a pairwise alignment could be recovered by any standard binary registration method. Unfortunately, that would require to solve a partial matching problem [8] between each *observation* shape and the *template*, which is far from trivial. Therefore we will sum equations for all shape domains $\mathcal{D}_i$ and solve the problem simultaneously, estimating all parameters in one system of equations. Thus Eq. (4) becomes

$$\sum_{i=1}^{\ell} \int_{\mathcal{D}_i} \omega_j(\mathbf{x})d\mathbf{x} = \sum_{i=1}^{\ell} |\mathbf{A}_i| \int_{\mathcal{D}'_i} \omega_j(\mathbf{A}_i\mathbf{y})d\mathbf{y} \ . \tag{5}$$

Let $\mathcal{D} := \cup_{i=1}^{\ell}\mathcal{D}_i$, where $\mathcal{D} = \{\mathbf{x} \in \mathbb{P}^n | \lambda(\mathbf{x}) \neq 0\}$ is the shape domain corresponding to the whole *template*. Therefore the left hand side of the above equation can be written as

$$\sum_{i=1}^{\ell} \int_{\mathcal{D}_i} \omega_j(\mathbf{x})d\mathbf{x} = \int_{\bigcup_{i=1}^{\ell}\mathcal{D}_i} \omega_j(\mathbf{x})d\mathbf{x} = \int_{\mathcal{D}} \omega_j(\mathbf{x})d\mathbf{x} \ , \tag{6}$$

which can be computed directly from the input image without knowing the partitioning $\mathcal{D}_i$. The resulting system of equations has $\ell n(n+1)$ unknowns:

$$\int_{\mathcal{D}} \omega_j(\mathbf{x})d\mathbf{x} = \sum_{i=1}^{\ell} |\mathbf{A}_i| \int_{\mathcal{D}'_i} \omega_j(\mathbf{A}_i\mathbf{y})d\mathbf{y}, \quad j = 1, \ldots, m \ . \tag{7}$$

The solution of the system Eq. (7) provides all the unknown parameters of the overall deformation. Since each ω_j provides one equation, we need $m \geq \ell n(n+1)$ linearly independent functions to solve for ℓ linear transformations. In practice, $m > \ell n(n+1)$ yielding an over-determined system for which a least squares solution is obtained.

3 Numerical Implementation

Theoretically, any nonlinear function satisfying Eq. (3) could be used to construct the system of equations Eq. (7). In practice, however, the solution is obtained via iterative least-squares minimization algorithms, like the *Levenberg-Marquardt algorithm* [17], requiring a carefully chosen numerical scheme.

3.1 Normalization

First of all, the coordinates of both images are normalized into the unit hypercube $[-0.5, 0.5]^n$, *i.e.* $\cup_{i=1}^{\ell} \mathcal{D}'_i \mapsto [-0.5, 0.5]^n$ and $\mathcal{D} \mapsto [-0.5, 0.5]^n$. This is achieved by translating the origin into the center of the mass of the *template* and *observation* followed by an appropriate isotropic scaling with a common factor corresponding to the maximum size of the *template* and *observation*. Of course, the solution of the nonlinear system has to be unnormalized to get the right transformations between the original shapes. Denoting the normalizing transformations of the *template* and *observation* by $\mathbf{N}_t$, $\mathbf{N}_o$, respectively and the solutions by $\widetilde{\mathbf{A}}_i$, the true transformation is thus obtained as $\widehat{\mathbf{A}}_i = \mathbf{N}_t^{-1} \widetilde{\mathbf{A}}_i \mathbf{N}_o$ for all $i = 1, \ldots, \ell$.

Since a least-squares solution involves minimizing the algebraic error of Eq. (7), we expect an equal contribution from each equation in order to guarantee an unbiased error measure. This is achieved by normalizing the range of each ω_j into $[-1, 1]$. We found experimentally, that the transformations occurring during the least-squares minimization process do not transform the shapes out of a hyper-sphere with center in the origin and a radius $\sqrt{n}/2$ (*i.e.* the circumscribed hyper-sphere of the unit hyper-cube). Thus the normalization can be done by dividing the integrals in Eq. (7) with an appropriate constant c_j corresponding to the maximal magnitude of the integral over this domain:

$$c_j = \int_{\|\mathbf{x}\| \leq \frac{\sqrt{n}}{2}} |\omega_j(\mathbf{x})| d\mathbf{x} \ , \quad j = 1, \ldots, m. \tag{8}$$

3.2 Algorithmic Solution and Complexity

In practice, only a limited precision digital image is available, thus the integrals can only be *approximated* by a discrete sum over the foreground pixels introducing an inherent, although negligible error into our computation. The continuous domains $\mathcal{D}$ and $\mathcal{D}'_i$ are represented as finite sets of foreground pixels denoted by D and D'_i. Thus the discrete form of the normalized Eq. (7) becomes

$$\frac{1}{c_j} \sum_{\mathbf{x} \in D} \omega_j(\mathbf{N}_t \mathbf{x}) = \frac{1}{c_j} \sum_{i=1}^{\ell} |\mathbf{A}_i| \sum_{\mathbf{y} \in D'_i} \omega_j(\mathbf{N}_o \mathbf{A}_i \mathbf{y}) \ , \quad j = 1, \ldots, m \ . \tag{9}$$

The system of Eq. (9) is solved by iterative least squares minimization using the *Levenberg-Marquardt algorithm* [17], which requires the evaluation of the equations at every iteration step. Thus the time complexity of the algorithm is considerably decreased if the sums can be precomputed, hence avoiding scanning the image pixels at every iteration. Theoretically, an arbitrary set of ω functions could be used, as long as they generate linearly independent equations. It is shown in [15], however, that choosing a set of polynomial functions will result in a polynomial system of equations, where these sums become precomputed constants. According to these findings the following set of polynomes are adopted

$$\{\omega_j : \mathbb{P}^n \to \mathbb{R}\}_{j=1}^{m} = \{\mathbf{x} \mapsto x_1^{u_1} \dots x_n^{u_n} | u_k \in \mathbb{N},\ k = 1, \dots, n,\ 0 \leq \sum_{k=1}^{n} u_k \leq d\}\ , \quad (10)$$

where d is the maximum degree and the number of the polynomes is given by $m = \frac{1}{n!} \prod_{i=1}^{n} (d + i)$.

The simple pseudo code of the algorithm is shown in Algorithm 1. Since a set of polynomial functions is applied to generate Eq. (9), the unknowns are eliminated from the sums [15]. Hence the algorithm has a linear time complexity: the complexity of constructing the system Eq. (9) is $\mathcal{O}(|D| + \sum_{i=1}^{\ell} |D'_i|)$; and the complexity of the solver itself is thus independent from the size of the input images.

Algorithm 1. Pseudo-code of the proposed algorithm.

Input : The binary *template* (D) and ℓ *observation* shapes (D'_i, $i = 1, \dots, \ell$)
Output: ℓ estimated linear transformations $\widehat{\mathbf{A}}_i$

1. Normalize the input coordinates by an appropriate similarity transformation $\mathbf{N}$ into $[-0.5, 0.5]^n$ such that the center of mass becomes the origin.
2. Choose a set of $\omega_j : \mathbb{P}^n \to \mathbb{R}$ ($j = 1, \dots, m \geq \ell n(n+1)$) polynomial functions.
3. Construct the (over-determined) system of equations Eq. (9).
4. Find a least-squares solution of the system using a *Levenberg-Marquardt* algorithm. The solver is initialized with the parameters of the identity transformation.
5. Unnormalizing the solutions $\widetilde{\mathbf{A}}_i$ gives the parameters of the aligning transformation as $\widehat{\mathbf{A}}_i = \mathbf{N}_t^{-1} \widetilde{\mathbf{A}}_i \mathbf{N}_o$.

4 Affine Transformations

Herein we apply the registration framework to important classes of linear deformations: 2D and 3D affine, and 3D rigid body. 2D affine transformations are often used as a linear approximation of projective distortions. 3D rigid body transformation is important in many medical applications. In particular, when bony structures need to be aligned in CT volumes then this transformation should be considered due to the bio-mechanical properties of bones.

4.1 2D Affine Transformations

A 2D affine transformation has 6 parameters, hence $n = 2$ and we have 6ℓ unknowns. In order to obtain sufficiently many equations by using the set of ω functions described in Eq. (10), d has to be chosen such that

$$m = \frac{(d+1)(d+2)}{2} \geq 6\ell \quad \Rightarrow \quad d \geq \left\lceil \frac{\sqrt{1+48\ell}-3}{2} \right\rceil , \tag{11}$$

where $\lceil \cdot \rceil$ denotes the upper integer parts. Eq. (7) becomes for all $j = 1, \ldots, m$

$$\int_{\mathcal{D}} x_1^{u_1} x_2^{u_2} d\mathbf{x} = \sum_{i=1}^{\ell} \int_{\mathcal{D}'_i} |\mathbf{A}_i| (a_{i11}y_1 + a_{i12}y_2 + a_{i13})^{u_1} (a_{i21}y_1 + a_{i22}y_2 + a_{i23})^{u_2} d\mathbf{y} \ , \tag{12}$$

where the Jacobian can be easily computed as $|\mathbf{A}_i| = a_{i11}a_{i22} - a_{i12}a_{i21}$.

4.2 3D Affine Transformations

The extension of the 2D case to 3D is rather straightforward. Here, the *template* parts undergo different 3D affine transformations, having a total of 12ℓ unknowns. In this case, d has to be chosen such that

$$m = \frac{(d+1)(d+2)(d+3)}{6} \geq 12\ell \quad \Rightarrow \quad d \geq \left\lceil \frac{c}{3} + \frac{1}{c} - 2 \right\rceil , \text{where}$$

$$c = \sqrt[3]{3\left(324\ell + \sqrt{(324\ell)^2 - 3}\right)}. \tag{13}$$

The Jacobian can be computed as in the 2D case.

4.3 3D Rigid-Body Transformations

An important special case of 3D linear deformations is the rigid-body transformation. This kind of transformations have six degree of freedom: $\alpha_1, \alpha_2, \alpha_3$ are the rotation angles and t_1, t_2, t_3 are the translations along the three coordinate axes. A similar set $\{\omega\}_{j=1}^{m}$ can be used as in Eq. (13), but we need fewer polynomes:

$$d \geq \left\lceil \frac{c}{3} + \frac{1}{c} - 2 \right\rceil , \text{ where } c = \sqrt[3]{3\left(27 + 162\ell + \sqrt{(27+162\ell)^2 - 3}\right)}. \tag{14}$$

Since a rigid-body transformation does not change the size of the objects, the Jacobian determinant equals to 1, hence it is omitted from the equations.

5 Experimental Results

The proposed method has been evaluated on 2D and 3D synthetic datasets. In the case of 2D transformations, the dataset consisted of 10 *template* objects. Synthetic observations were generated by first cutting each object into 2 parts in 4 different ways, resulting in 4 images for each *template*. Then 600 *observations* of size 700×700 were generated by applying randomly composed affine transformations to each of these images with the following parameter ranges: rotation angles of $[-\pi/4; \pi/4]$ and along both axes scaling factors from $[0.75; 1.25]$, skewing from $[-0.1; 0.1]$, and translations of $[-25; 25]$. In the 3D case, 10 *template* volumes were randomly cut into 2 parts by a plane, such that the smaller part is at least 20% of the original volume. By cutting each volume in five different ways, 50 volume images are obtained. Then random 3D affine transformations with similar parameters as in the 2D case (the only difference is that translations were chosen from [-10;10]) have been used to generate a total of 200 3D *observations* of size $250 \times 250 \times 250$.

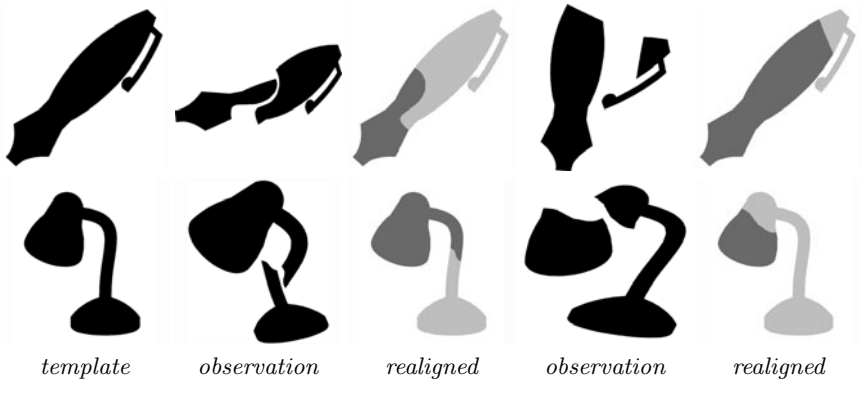


Fig. 1. Sample results on 2D synthetic images

For the evaluation of the results, we defined two kind of error measures: The first one (denoted by ϵ) measures the average distance between the true $\mathbf{A}_i$ and the estimated $\widehat{\mathbf{A}}_i$ transformation for all object. The second one is the absolute difference (denoted by δ) between the *template* and the *aligned* shapes:

$$\epsilon = \sum_{\mathbf{p} \in D'_i, 1 \le i \le \ell} \frac{\|(\mathbf{A}_i - \widehat{\mathbf{A}}_i)\mathbf{p}\|}{|D'|}, \quad \text{and} \quad \delta = \frac{|\widehat{D} \triangle D|}{|\widehat{D}| + |D|} \cdot 100\%, \tag{15}$$

where $\triangle$ means the symmetric difference, while $D' = \cup_{i=1}^{\ell} D'_i$ and $\widehat{D} = \cup_{i=1}^{\ell} \widehat{D}_i$ denote the set of pixels of the *observation* and *aligned* shape respectively. Intuitively, ϵ shows the average transformation error per pixel. Note that ϵ can only

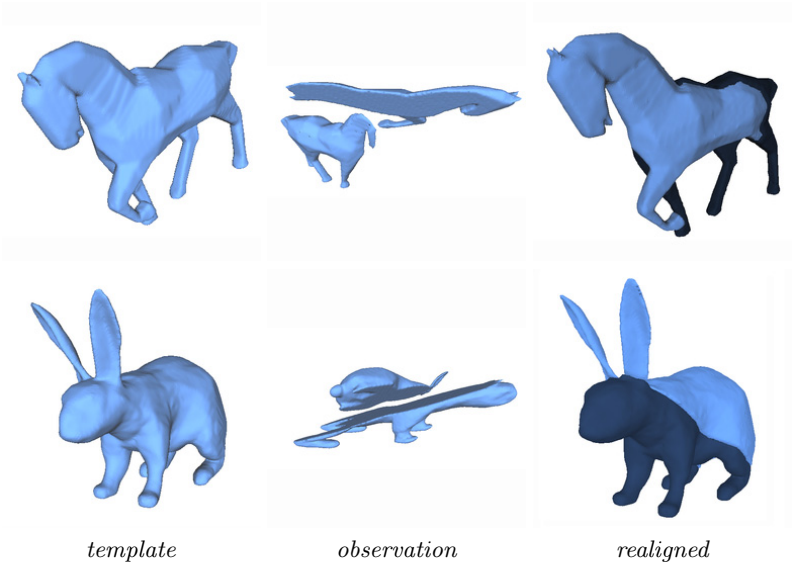


Fig. 2. Sample results on 3D synthetic images

be used when the true transformation is also known, while δ can always be computed. On the other hand, ϵ gives a better characterization of the transformation error as it directly evaluates the mistransformation. As a subjective evaluation measure, we found experimentally that a $\delta \leq 5\%$ in 2D and a $\delta \leq 10\%$ in 3D corresponds to a visually good alignment.

The proposed method was implemented in Matlab and ran under Linux with 3GHz CPU and 3GB memory. The typical runtime was under 3 seconds for 2D and 10 seconds for 3D shapes. Some results are shown in Fig. 1 and Fig. 2. Quantitative results in Table 1 clearly show that the proposed method provides almost perfect alignments in both 2D and 3D.

5.1 Robustness

In practice, segmentation never produces perfect shapes. Therefore, besides using various kind of real images inherently subject to such errors, we have also evaluated the robustness of the proposed approach against different type of segmentation errors. In the first testcase, $5\%, \ldots, 20\%$ of the foreground pixels has been removed from the *observation* before registration. In the second case, we occluded continuous square-shaped regions of size equal to $1\%, \ldots, 10\%$ of the shape. Finally, we randomly added or removed squares uniformly around the boundary of a total size $1\%, \ldots, 10\%$ of the shape. Note that we do not include cases where erroneous foreground regions appear as disconnected regions, because such false regions can be efficiently removed by appropriate morphological filtering. We therefore concentrate on cases where segmentation errors cannot be

(a) missing pixels (b) occlusion (c) boundary error

Fig. 3. Sample observations with various degradations

Table 1. Median of error measures achieved by the proposed method on the 2D and 3D synthetic datasets. The first two rows show the results without degradation while the rest contains the error values vs. various type of segmentation errors as shown in Fig. 3.

Without degradation	ϵ (pixel)				δ (%)			
2D affine transformations	0.11				0.13			
3D affine transformations	0.7				3.09			
(a) missing pixels	1%	5%	10%	20%	1%	5%	10%	20%
2D affine transformations	6.57	21.1	32.83	56.26	2.09	6.24	8.39	12.62
3D affine transformations	1.22	4.65	9.71	19.02	3.99	8.67	15.8	23.54
(b) size of occlusion	1%	2.5%	5%	10%	1%	2.5%	5%	10%
2D affine transformations	9.91	20.45	35.04	58.68	3.54	6.35	9.51	13.75
3D affine transformations	3.27	7.7	14.73	22.74	8.07	13.08	18.47	26.13
(c) size of boundary error	1%	2.5%	5%	10%	1%	2.5%	5%	10%
2D affine transformations	1.9	3.91	6.65	12.23	0.59	1	1.73	3.08
3D affine transformations	0.99	1.44	2.33	4.03	3.23	3.65	4.44	5.8

filtered out. See samples of these errors in Fig. 3. Table 1 shows that our method is quite robust whenever errors are uniformly distributed over the whole shape (first and third testcases). However, it becomes less stable in case of larger localized errors, like occlusion and disocclusion. This is a usual behavior of area-based methods because they are relying on quantities obtained by integrating over the object area. Thus large missing parts would drastically change these quantities resulting in false alignments. Nevertheless, in many application areas one can take images under controlled conditions which guarantees that observations are not occluded (*e.g.* medical imaging, industrial inspection).

5.2 Solving the Tangram Puzzle

Tangram is a dissection puzzle consisting of seven flat tiles (called *tans*), which are put together to form various shapes. The objective is to form a specific shape given only by its silhouette. Fig. 4 shows some examples of these shapes and the solutions found by our method.

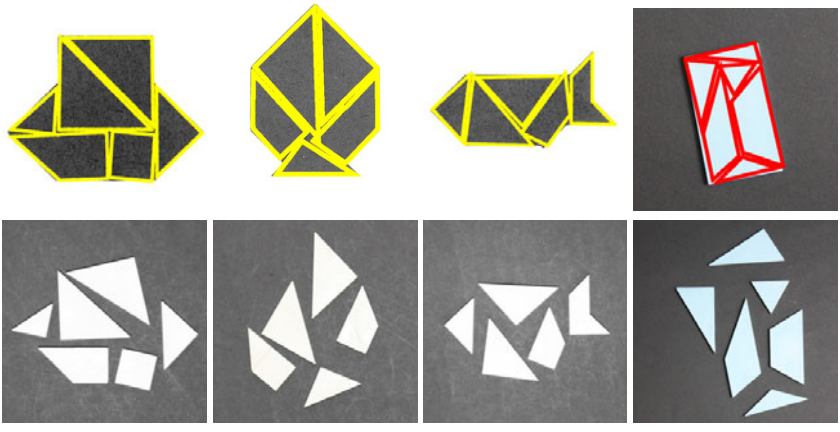


Fig. 4. Solutions of the Tangram puzzle. **Top:** Template images with overlayed contours of aligned fragments. **Bottom:** Observations.

The images were taken with a digital camera, then they were thresholded and the resulting 2D shapes were realigned according to the *template*. The first three *templates* of Fig. 4 are more challenging as they are scanned versions of the printed shapes found in the Tangram manual, which are only approximate silhouettes of the final tile configurations. We have used the affine model as an approximation of the actual plane projective transformation acting between the shapes.

It is well known that the *Levenberg-Marquardt* algorithm finds a local minima close to the initialization. Finding a good initial configuration is largely application-dependent. For example, on these images a global optimization procedure (*e.g.* Spectral Gradient Method [18]) provided good initialization, from which *Levenberg-Marquardt* gives a better solution than starting from the identity transform.

Finally, we note that some tiles are slightly overlapping in Fig. 4. This is because overlaps are invisible for the system of equations. Nevertheless, overlaps could be prevented by checking the transformed fragments at every iterations, but this is a rather time consuming procedure.

5.3 Realigning Bone Fractures

Complex bone fracture reduction frequently requires surgical care, especially when angulation or displacement of bone fragments are large. In such situations, computer aided surgical planning [5] is done before the actual surgery takes place, which allows to gather more information about the dislocation of the fragments and to arrange and analyze the surgical implants to be inserted. A crucial part of such a system is the relocation of bone fragments to their original anatomic position. Since the input data is typically a volume CT image, this

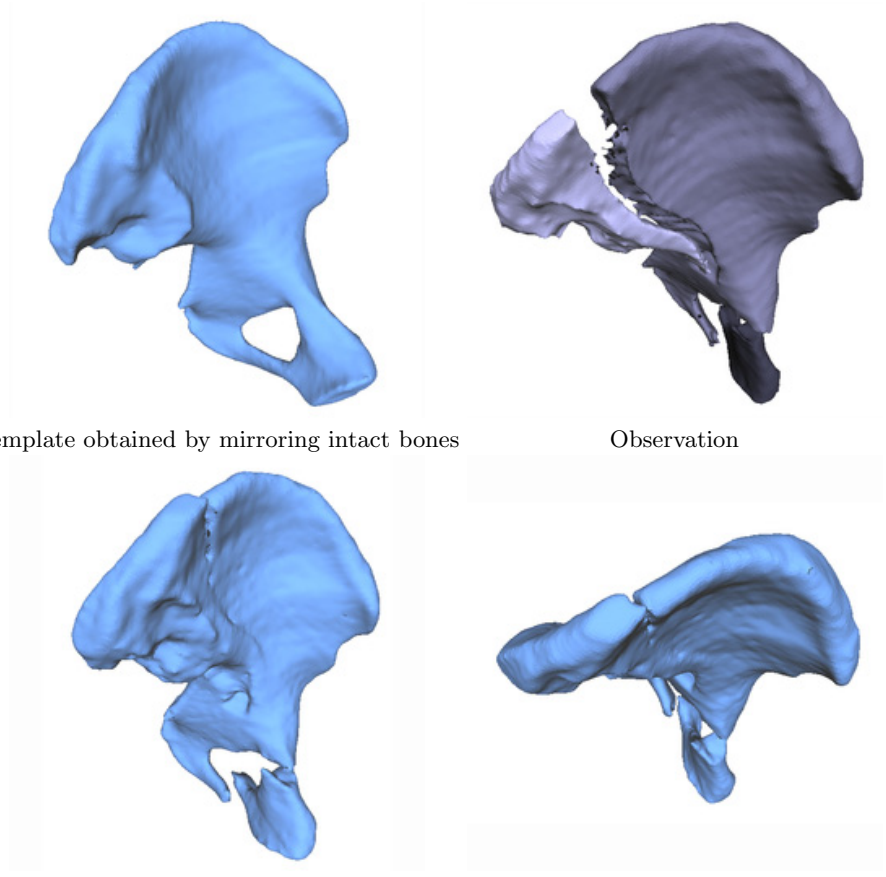


Fig. 5. Bone fracture reduction (CPU time was 15 sec. for these 1 megavoxel CT volumes)

repositioning has to be performed in 3D space which requires an expensive special 3D haptic device and quite a lot of manual work. Therefore automatic bone fracture reduction can save considerable time, providing experts with a rough alignment which can be manually fine-tuned according to anatomic requirements.

Since surgical planning involves the biomechanical analysis of the bone with implants, only rigid-body transformations are allowed. In [5], a classical ICP algorithm is used to realign fractures. Winkelbach *et al.* [6] proposed an approach for estimating the relative transformations between fragments of a broken cylindrical structure by using well known surface registration techniques, like 2D depth correlation and the ICP algorithm. In [7], registration is solved by using quadrature filter phase difference to estimate local displacements.

Herein, we apply our puzzle framework to reduce pelvic fractures using 3D rigid-body transformations. In cases of single side fractures, the *template* is

simply obtained by mirroring intact bones of the patient. Fig. 5 shows a typical result for a pelvic fracture with three fragments. The main challenges are segmentation errors and, due to the variability of the human body, a slightly different *template*. In spite of these difficulties, the alignment of larger parts is quite accurate, only the small fragment has a noticeable alignment error. Since the error caused by a misplaced small piece is relatively low, the solver may not find the best transformation. If we could normalize the terms of Eq. (9) corresponding to each fragment, then the algebraic error would be better balanced and a precise alignment could be found. Unfortunately, this is impossible as we should know the partitioning of the *template* to compute proper normalizing constants. Since human verification and correction of the result is needed anyway in a real surgical planning system, these small errors are not critical and can be easily corrected.

6 Conclusion

A novel framework to solve the affine puzzle problem has been proposed and applied to 2D and 3D affine transformations. As opposed to classical solutions based on landmark extraction and correspondences, the proposed solution finds the aligning transformations without any additional information. Basically, the method consists in constructing a polynomial system of equations whose solution directly provides the unknown parameters. Obviously, the number of object fragments and strength of the deformation may influence the quality of the alignment: The more parts we have the more equations are required, which affects numerical stability. Furthermore, more parts allow more affine transformations yielding a stronger deformation. A completely random fragment-configuration corresponds to a complex deformation, for which a stable solution is difficult to achieve. On the other hand, when pieces are in relative order then a rather accurate solution is obtained. Note, that for the presented medical application, this is a realistic assumption due to physical constraints. Quantitative evaluations on both 2D and 3D synthetic datasets demonstrate the performance and robustness of the method and results obtained on real images confirm its relevance in various application domains.

References

1. Kong, W., Kimia, B.B.: On solving 2D and 3D puzzles using curve matching. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, vol. 2, pp. 1–8. IEEE, Los Alamitos (2001)
2. Papaioannou, G., Karabassi, E.A., Theoharis, T.: Reconstruction of three-dimensional objects through matching of their parts. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 114–124 (2002)
3. McBride, J.C., Kimia, B.B.: Archaeological fragment reconstruction using curve-matching. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition Workshop, Madison, Wisconsin, USA, pp. 1–8. IEEE, Los Alamitos (2003)

4. Hill, D.L.G., Batchelor, P.G., Holden, M., Hawkes, D.J.: Medical image registration. Physics in Medicine and Biology 45, R1–R45 (2001)
5. Erdőhelyi, B., Varga, E.: Semi-automatic bone fracture reduction in surgical planning. In: Proceedings of the International Conference on Computer Assisted Radiology and Surgery. International Journal of Computer Assisted Radiology and Surgery, vol. 4, pp. 98–99. Springer, Berlin (2009)
6. Winkelbach, S., Westphal, R., Goesling, T.: Pose estimation of cylindrical fragments for semi-automatic bone fracture reduction. In: Michaelis, B., Krell, G. (eds.) DAGM 2003. LNCS, vol. 2781, pp. 556–573. Springer, Heidelberg (2003)
7. Pettersson, J., Knutsson, H., Borga, M.: Non-rigid registration for automatic fracture segmentation. In: Proceedings of International Conference on Image Processing, Atlanta, GA, USA, pp. 1185–1188. IEEE, Los Alamitos (2006)
8. Bronstein, A.M., Bronstein, M.M.: Regularized partial matching of rigid shapes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 143–154. Springer, Heidelberg (2008)
9. Funkhouser, T., Shilane, P.: Partial matching of 3D shapes with priority-driven search. In: Proceedings of the Eurographics Symposium on Geometry Processing, Sardinia, Italy, Eurographics, ACM SIGGRAPH, pp. 1–12 (2006)
10. Reuter, M.: Hierarchical shape segmentation and registration via topological features of Laplace-Beltrami eigenfunctions. International Journal of Computer Vision 89, 287–308 (2010)
11. Rustamov, R.M.: Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In: Proceedings of the Eurographics Symposium on Geometry Processing, Barcelona, Spain, Eurographics, ACM SIGGRAPH, pp. 1–9 (2007)
12. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14, 239–256 (1992)
13. Periaswamy, S., Farid, H.: Medical image registration with partial data. Medical Image Analysis 10, 452–464 (2006)
14. Feldmar, J., Ayache, N.: Rigid, affine and locally affine registration of free-form surfaces. International Journal of Computer Vision 18, 99–119 (1996)
15. Domokos, C., Kato, Z.: Parametric estimation of affine deformations of planar shapes. Pattern Recognition 43, 569–578 (2010)
16. Hagege, R.R., Francos, J.M.: Estimation of affine geometric transformations of several objects from global measurements. In: Proceedings of IEEE International Workshop on Multimedia Signal Processing, Rio de Janeiro, Brazil, pp. 1–5. IEEE, Los Alamitos (2009)
17. Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. SIAM Journal on Applied Mathematics 11, 431–441 (1963)
18. Birgin, E.G., Martínes, J.M., Raydan, M.: SPG: Software for convex-constrained optimization. ACM Transactions on Mathematical Software 27, 340–349 (2001)

Location Recognition Using Prioritized Feature Matching*

Yunpeng Li, Noah Snavely, and Daniel P. Huttenlocher

Department of Computer Science, Cornell University, Ithaca, NY 14853
{yuli,snavely,dph}@cs.cornell.edu

Abstract. We present a fast, simple location recognition and image localization method that leverages feature correspondence and geometry estimated from large Internet photo collections. Such recovered structure contains a significant amount of useful information about images and image features that is not available when considering images in isolation. For instance, we can predict which views will be the most common, which feature points in a scene are most reliable, and which features in the scene tend to co-occur in the same image. Based on this information, we devise an adaptive, prioritized algorithm for matching a representative set of SIFT features covering a large scene to a query image for efficient localization. Our approach is based on considering features in the scene database, and matching them to query image features, as opposed to more conventional methods that match image features to visual words or database features. We find this approach results in improved performance, due to the richer knowledge of characteristics of the database features compared to query image features. We present experiments on two large city-scale photo collections, showing that our algorithm compares favorably to image retrieval-style approaches to location recognition.

Keywords: Location recognition, image registration, image matching, structure from motion.

1 Introduction

In the past few years, the massive collections of imagery on the Internet have inspired a wave of work on location recognition—the problem of determining where a photo was taken by comparing it to a database of images of previously seen locations. Part of the recent excitement in this area is due to the vast number of images now at our disposal: imagine building a world-scale location recognition engine from all of the geotagged images from online photo collections, such as Flickr and street view databases from Google and Microsoft. Much of this recent work has posed the problem as that of image retrieval [1,2,3,4]: given a query image to be recognized, find a set of similar images from a database using image descriptors or visual features (possibly with a geometric verification step), often building on bag-of-words techniques [5,6,7,8,9]. In this type of approach, the database images are largely treated as independent collections of features, and any structure between the images is ignored. In this paper we consider exploiting this potentially rich structure for use in location recognition.

* This work was supported in part by NSF grant IIS-0713185, Intel, Microsoft, and Google.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 791–804, 2010.

For instance, recent work has shown that it is possible to automatically estimate correspondence information and reconstruct 3D geometry from large, unordered collections of Internet images of landmarks and cities [10,3,11]. Starting with such structure, rather than a collection of raw images, as a representation for location recognition and localization is promising for several reasons. First, the point cloud is a compact "summary" of the scene—it typically contains orders of magnitude fewer points than there are features in an image set, in part because each 3D point represents a cluster of related features, but also because many features extracted by algorithms like SIFT are noisy and not useful for matching. Second, for each reconstructed scene point we know the set of views in which a corresponding image feature was detected, and the size of this set is related to the "stability" of that point, i.e., how reliably it can be detected in an image, as well as how visible that scene feature is (e.g., a point on a tower might be more visible than other features, see Figure 1). We can also determine how correlated two points are—i.e., how often they are detected in the same images. Finally, when using Internet photo collections to build our reconstruction, the number of times a point is viewed is related to the "popularity" of a given viewpoint—some parts of a scene may be photographed much more often than others [12].

Fig. 1. SIFT Features in an image corresponding to reconstructed 3D points in the full model (left) and the compressed model (right) for Dubrovnik. The feature corresponding to the most visible point (i.e., seen by the most number of images) is marked in red in the right-hand image. This feature, the face of a clocktower, is intuitively a highly visible one, and was successfully matched in 370 images (over 5% of the total database).

In this paper, we explore how these aspects of reconstructed photo collections can be used to improve location recognition. In particular, we use *scene features* (corresponding to reconstructed 3D points), rather than images, as a matching primitive, revisiting nearest-neighbor feature matching for this task. While there is a history of matching individual features for recognition and localization [13,14,15], we advocate reversing the usual process of matching. Typically, image features are matched to a database of features. Instead, we match database features to image features, motivated by the richer information available about scene features relative to query image features. Towards this end, we propose a new, prioritized point matching algorithm that matches a subset

of scene features to features in the query image, ordered by our estimate of how likely a scene feature is to be matched given our prior knowledge about the database points as well as which scene features have been successfully matched so far. This prioritized approach allows "common" views to be localized quickly, sometimes with just a few hundred nearest neighbor queries, even for large 3D models. In our experiments this approach also successfully localizes a higher proportion of images than approaches based on image retrieval. In addition, we show that compressing the model by keeping only a subset of representative points is beneficial in terms of both speed and accuracy. We demonstrate results on large Internet databases of city images.

Given the feature correspondences found using our algorithm, we next estimate the exact pose of the camera. While this final step relies on having an explicit 3D reconstruction, many of the ideas used in our approach—finding stable points, prioritizing features to match, etc.— only require knowledge of correspondences between features across the image database ("feature tracks"), and not 3D geometry per se. However, because we ultimately produce a camera pose, and because the global geometry imposes additional consistency constraints on the correspondences, we represent our scene with explicit geometry, and refer to our database of scene features as a "point cloud."

2 Related Work

Our work is most closely related to that of Irschara et al. [4], which also uses SfM point clouds as the basis of a location recognition system. Our work differs from theirs in several key ways, however. First, while they use a point cloud to summarize a location, their recognition algorithm is still based on an initial image retrieval step using vocabulary trees. In their case, they generate a minimal set of "synthetic" images that covers the point cloud, and, given a new query image, use a vocabulary tree to retrieve similar images in this covering. In one sense, our approach is the dual of [4]: instead of selecting a set of images that cover the 3D points, we select a minimal set of 3D points that cover the images, and use these points themselves as the matching primitives. Second, while [4] uses images taken by a single person, we use city-scale image databases taken from the Internet. Such Internet databases differ from more structured datasets in that have much wider variation in appearance, and also reflect the inherent "popularity" of given views. Our approach is sensitive to and exploits both of these properties.

Our work is also related to the city-scale location recognition work of Schindler et al. [1], who also use image feature stability, as well as the *distinctiveness* features, as cues for building a recognition database. As with [4], Schindler et al.use image retrieval as a basis for location recognition, and use a database of images taken a regular samples throughout a city.

Like [4], [1], and [15], part of our approach involves reducing the amount of data used to represent locations, a theme which has been explored by other researchers as well. For instance, [16] uses epitomes [17] as compact representations of locations created from videos of different scenes. Li et al. [3] derive "iconic" images derived from performing clustering on large Internet photo collections, then localize query images by retrieving similar iconic images using bag-of-words or GIST descriptors [18].

Similarly, [19] builds a landmark recognition engine by selecting iconic images using a graph-based algorithm.

Finally, a number of researchers have applied more traditional recognition and machine learning techniques the problem of location recognition [2, 20, 21]. Others have made use of information from image sequences; while this is a common approach in the SLAM (Simultaneous Localization and Mapping) community, human travel priors have also been used to georegister personal photo collections [22].

3 Building a Compact Model

Given a large collection of images of a specific area of interest (e.g., "Rome") downloaded from the Internet, we first reconstruct one or more 3D models using image matching and structure from motion (SfM) techniques. Our reconstruction system is based on the work of Agarwal et al. [23]; we use a vocabulary tree to propose an initial set of matching image pairs, do detailed SIFT feature matching to find feature correspondences between images, then use SfM to reconstruct 3D geometry. Because Internet photos taken in cities tend to have high density near landmarks or other popular places, and low density elsewhere, a set of city photos often breaks up into several connected components (and a large number of "singleton" images that are not matched to any other photo—we remove these from consideration, as well as other very small connected components). For instance, the Rome dataset described in Section 5 consists of 69 large components. An example 3D reconstruction is shown in Figure 2. Each reconstruction consists of a set of recovered camera locations, as well as a set of reconstructed 3D points, denoted P. For each point $p \in P$, we know the set of images in which p was successfully detected and matched during the feature matching process (and deemed to be a geometrically consistent detection). We also have a 128-byte SIFT descriptor for each detection (we will assign their mean descriptor to p). Given a new query image from the same scene, our goal is to find correspondences with these "scene features," and determine the camera pose.

One property of Internet photo collections (and current SfM methods) is that there is a large variability in the number of times each scene feature is matched between images. While many scene points are matched in only two images, others might be successfully matched in hundreds. Consequently, not all scene features are equally useful

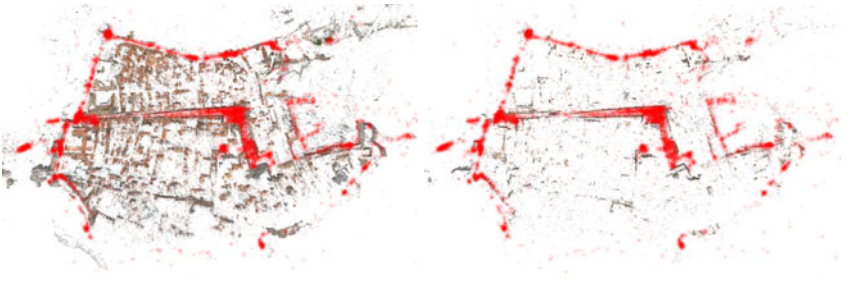


Fig. 2. Reconstructed 3D point cloud for Dubrovnik. Left: full model. Right: compressed model (P^c). The bright red regions represent the distribution of reconstructed camera positions.

when matching with a new query image. This suggests a first step of "compressing" the set of scene features by keeping only a subset of informative points, reducing computational cost and suppressing potential sources of confusion.

A naïve way to compress the model is to rank the points by "visibility" (i.e., the number of images where that point has been detected and matched) and select a set from the top of the list. However, points selected in such way can (and usually do) have very uneven spatial distribution, with popular areas having a large number of points, and other areas having few or none.

Instead, we would like to choose a set of points that are both prominent and that cover the whole model. To this end, we pose the selection of points as a set covering problem, where the images in the model are the elements to be covered and each point is regarded as a set containing the images in which it is visible. In other words, we seek the smallest subset of P, such that each image is covered by at least one point in the subset. Given such a subset, we might expect that a query image drawn from the same distribution of views as the database images would—roughly speaking—match at least one point in the model. However, because feature matching is a noisy process, and because robust pose estimation requires more than one a single match, we instead require that the subset covers each image at least K times (e.g., where $K = 100$).[1] Although computing the minimum set K-cover is NP-hard, a good approximation can be found using a greedy algorithm that always selects the point which covers the largest number of not-yet-fully covered images. Note that this is different from the covering problem in [4], which aims to cover the points instead of the images. Our covering criterion is also related to the *informative features* used in Schindler et al. [1], though our method is different; we choose features based on explicit correspondences from feature matching and SfM, and do not use an explicit measure of feature distinctiveness.

For our problem, given our initial point set P, we compute two different K-covers: one for $K = 5$ but limited to 2,000 points (via early stopping of the greedy algorithm), denoted P^s, and one for $K = 100$ (without explicit limit on the number of points), denoted P^c. Intuitively, P^s forms an extremely compact "seed" description of the entire model that can be quickly swept through to find promising areas in which to focus the search for further feature matches, while P^c is a more exhaustive representation that can be used for more accurate localization. We describe how our matching algorithm uses these sets in the next section.

In our experiments, the reduction of points from model compression is significant. For the Dubrovnik set, the point set was reduced from over 2 million in the full model to under 80,000 in the compressed model P^c. Figure 1 shows the features corresponding to reconstructed 3D points in the full and the compressed Dubrovnik models that are visible in one particular image. The point clouds corresponding to the full and compressed models are shown in Figure 2.

4 Registration and Localization

The ultimate goal of our system is to produce an accurate pose estimate of a query image, given a relevant database of images. To register a query image to the 3D model

[1] If an image sees fewer than K points in the full model, all of those points are selected.

using pose estimation, we need to first find a set of correspondences between scene features in the model and image features in the query image. While many recent approaches initially pose this as an image retrieval problem, we reconsider an approach based purely on directly finding correspondences using nearest-neighbor feature matching. In our case, we represent each point in our model with the mean of the corresponding image feature descriptors. While the mean may not necessarily be representative of clusters of SIFT features that are large or multi-modal, it is a commonly used approach for creating compact representations (e.g. in k-means clustering) and we have found this simple approach to work well in practice (though better cluster representations is an interesting area for future work).

Given a set of SIFT descriptors in a query image (in what follows we refer to these as "features," for simplicity) and a set of SIFT descriptors representing the points in our model (we will refer to these descriptors as "points"), we consider two basic matching strategies:

- *Feature-to-point matching*, or *F2P*, where one takes each feature (in the query image), and finds the best matching point (in the database) and
- *Point-to-feature matching*, or *P2F*, where one conversely matches points in the model to features in the query image.

At first glance, F2P matching seems like the natural choice, since we usually think of matching a query to a database—not vice versa—and even the compressed model is usually much larger than the set of features in a single image. However, P2F matching has the desirable property that we have a significant amount of information about how informative each database point is, and which database points are likely to appear together, while we do not necessarily have much prior information about the features in an image (other than low-level confidence measures provided by the feature detector). In fact, counter to intuition, we show that P2F matching can be made to find matches more quickly than F2P—especially for popular images—by choosing an intelligent priority ordering of the points in the database, such that we often only need to consider a relatively small subset of the model points before sufficient matches are found. We evaluate the empirical performance of F2P and P2F matching in Section 5.

For both matching strategies, we find potential matches using the approximate nearest neighbor search [24]. As in prior work, we found that the priority search algorithm worked well in pratice. We used a fixed search limit of 250 points per query; increasing the search limit did not lead to significantly better results in our experiments.

We use a modified version of the ratio test that is common in nearest neighbor matching to classify a match as true or false. For P2F matching, we use the usual ratio test: a match between a point p from the model and feature f from the query image is considered a true match if $\frac{dist(p,f)}{dist(p,f')} < \lambda$, where $dist(\cdot,\cdot)$ is the distance between the corresponding SIFT descriptors, f' is the second nearest neighbor of p among features in the query image, and λ is the threshold ratio (0.7 in our implementation). For F2P matching, we found that the analogous ratio test does not perform as well. We speculate that this might be because the number of points in the compressed model is larger (and hence denser) than the typical image feature set, and this increased density in SIFT space has an effect on the ratio test. Instead, given a feature f and its approximate

nearest neighbor p in the model, we compute the second nearest neighbor f' to p in the image, and threshold on the ratio $\frac{dist(f,p)}{dist(f',p)}$ (note that this ratio could be ≥ 1). We found this test to perform better for F2P matching.

For P2F matching, we could find correspondences by applying the above procedure to *every* point in the model. However, this would be very costly. We can do much better by prioritizing certain queries over others, as we describe next.

4.1 Prioritized Point-to-Feature (P2F) Matching

As noted earlier, each point in the reconstructed 3D model is associated with the set of images in which it is successfully detected and matched ("visible"). Topologically, the model can be regarded as a bipartite graph (which we call the *visibility graph*) where each point and each image is a vertex, and where the edges encode the visibility relation.

Based on this relation, we develop a matching algorithm guided by three heuristics:

1. Points with higher degree in the visibility graph should generally be considered before points with lower degree, since a highly visible point is intuitively more likely to be visible in a query image. Our matching algorithm thus maintains a *priority* for each point, initially equal to its degree in the visibility graph.
2. If two points are often visible in the same set of images, and one of them has been matched to some feature in the query image, then the other should be more likely to find a match as well.
3. The algorithm should be able to quickly "explore" the model before "exploiting" the first two heuristics, so as to avoid being trapped in some part that is irrelevant to the query image. To this end, a set of highly visible *seed points* (corresponding to P^s in Section 3) are selected as a preprocess; these seed points are the basis for an initial exploratory round of matching (before moving on to the more exhaustive P^c model). We limit the size of P^s to 2,000 points, which is somewhat arbitrary but represents a balance between exploration and exploitation.

Our P2F matching algorithm matches model points to query features in priority order (using a priority queue), always choosing the point with the highest priority as the next candidate for matching. The priorities of all points are initially set to be proportional to their degrees in the visibility graph, i.e. $d_i = \sum_j V_{ij}$. The priorities of the "seed" points are further increased by a sufficiently large constant, so that all seed points rank above the rest. Whenever the algorithm finds a match to a point p that passes the ratio test, it increases the priority of related points, i.e., points seen in the same images as p. Thus, if the algorithm finds a true match, it can quickly home in on the correct part of the model and find additional true matches. If a true match is found early on (especially likely with a popular image), the image can be localized with a relatively small number of comparisons.

The matching algorithm terminates (successfully) once it has found a sufficient number of matches (given by a parameter N), or (unsuccessfully) when it has tried to match more than $500N$ points. On success, the matches are passed onto the pose estimation routine. The abort criterion is based on our empirically observed "background" match rate of roughly $1/500$, i.e., in expectation every one out 500 points will succeed the ratio test and be matched to some feature purely by chance (see also Section 5). Hence

when $500N$ points have been tried, the rate of finding matches so far is no higher than the background rate and hence a strong indication that the number of outlier matches will be very large and hence the image will most likely not be recognized by the model. The algorithm also depends on a second parameter ω, which is the trade-off between

Algorithm 1. Prioritized P2F Matching

Input: set of seed points P^s and compressed points P^c, n-by-m visibility matrix V, a query image
Output: Set of matches M
Parameters: Number of uniquely matched features required N, weight factor ω

$M, Y \leftarrow \emptyset$ (* Initialize the set of matches M and uniquely matched features Y *)
For all i ($i = 1 \cdots n$), $S_i \leftarrow d_i$, where $d_i = \sum_j V_{ij}$ (* Initialize priorities S *)
For all $i \in P$, S_i is incremented by a sufficiently large constant
$t \leftarrow 0$

while $\max S > 0$ and $|Y| < N$ **do**
  $i \leftarrow \arg\max S$, $t \leftarrow t + 1$
  Search for an admissible match among the features in the query image for X_i
  if such a feature y is found **do**
    $M \leftarrow M \cup \{(X_i, y)\}$, and $Y \leftarrow Y \cup \{y\}$
    for each j, s.t. $V_{ij} = 1$ **do** (* Update the priorities *)
      for each k, s.t. $V_{kj} = 1$ **do**
        $S_k \leftarrow S_k + \omega / d_i$
      end for
    end for
  end if
  $S_i \leftarrow -\infty$
  If $t \geq 500N$, abort
end while

the static (first) heuristic and dynamic (second) heuristic. A higher value of ω makes the priority of a point depend more on how well nearby points (in the visibility graph) have been matched, and less on its inherent visibility (i.e. its degree); a zero value for ω, on the other hand, would disable dynamic priorities altogether. We set $\omega = 10$, which heavily favors the dynamic priorities.

Our full algorithm for prioritized point-to-feature matching is given in Algorithm 1. We use a value $N = 100$, which appears to be sufficient for subsequent localization. In the case that the output M contains duplicates, i.e., multiple points are matched to the same feature, only the closest match (in terms of the distance between their SIFT descriptors) is kept.

Although the update of priorities that corresponds to the two nested inner loops of Algorithm 1 may seem to be a significant extra computational cost, these updates only occur when a match is found. The overhead is further reduced by updating the priorities only at fixed intervals, i.e., after every certain number of iterations (100 in our implementation). This also allows the algorithm to be conveniently parallelized or ported to a GPU, though we have not yet implemented these enhancements.

4.2 Feature-to-Point (F2P) Matching

For feature-to-point (F2P) matching, it is not clear if any ordering of image features is better than another. Therefore all features in the query image are considered for matching. In our experiments, we find that not considering all features always decreases the registration performance of F2P matching in our experiments.

4.3 Camera Pose Estimation

If the matching algorithm terminates successfully, then the set of matches M links 2D features in the query image directly to 3D points in the model. These matches are fed directly into our pose estimation routine. We use the 6-point DLT approach to solve for the projection matrix of the query camera, followed by a local bundle adjustment to refine the pose.

5 Experiments

We evaluated the performance of our method on three different image collections. Two of the models, Dubrovnik and Rome, were built from Internet photos retrieved from Flickr; the third, Vienna, was built from photos taken by a single calibrated camera (the same dataset used in [4]). Figure 2 shows part of the reconstructed 3D model for Dubrovnik and Rome. For each dataset, the number of registered images was in the thousands, and the number of 3D points in the full model in the millions; statistics are shown in Table 1.

Table 1. Statistics for each 3D model. Each row lists the name of the model, the number of cameras used in its construction, the number of points, and number of connected components in the reconstruction.

Model	# Cameras	# Points	# CCs
Dubrovnik	6844	2,208,645	1
Rome	16,179	4,312,820	69
Vienna	1,324	1,132,745	3

In order to obtain relevant test images (i.e. images that *can* be registered) for Dubrovnik and Rome, we first built initial models using all available images. A random subset of the images that were accepted by the 3D reconstruction process was then removed from these initial models and withheld as test images. This was done by removing their contribution to the SIFT descriptors of any points they see, as well as deleting any points that are no longer visible in at least two images. The resulting model no longer has any information about the test images, while we can still use the initial camera poses as "ground truth." For Dubrovnik and Rome, we also included the relevant test images *of the other* data set as negative examples. In all our experiments no irrelevant images were falsely registered. For Vienna, the set of test images are the same Internet photos as used in [4] (these images were not used in building the model). In all cases, the test images are downsampled to a maximum of 1600 pixels in both width and height.

The Vienna data set is different from Dubrovnik and Rome in that it is taken by the same camera over a short period of time, and is much more controlled, uniformly sampled, and homogeneous in appearance. Thus, it does not necessarily generalize as well to diverse query images, such as the Internet photos used in the test set (e.g, the stability of a point in the model may not necessarily be a good predictor of its stability in an arbitrary query image). Indeed, we found that using a model compressed with $K = 100$ for this collection was not adequate, likely because a covering for each image in the model may not also cover a wide variety of images with different appearance. Hence we used a larger covering ($K = 1000$) for this data set. Other than this, all parameters of the our algorithm are the same throughout the experiments.

As an extra validation step, we also created a second model for Vienna in the same way as we did for Dubrovnik and Rome, first building an initial model including all images, then removing the test images. We found that the model created in this way performed no better than the one built without ever involving the test images. This suggests that our evaluation methodology for Dubrovnik and Rome does not favorably bias the the results.

Table 2. Results for Dubrovnik. The test set consists of 800 relevant images and 1000 irrelevant ones (from Rome).

		Images	NN queries by P2F		Time in seconds	
		registered	registered	rejected	registered	rejected
Compressed model	P2F	753	9756	46433	0.73	2.70
(76645 points)	F2P	667	-	-	1.62	1.62
	Combined	753	-	-	0.70	3.96
	Seedless P2F	747	9986	46332	0.75	2.69
	Static P2F	693	16722	46558	1.11	2.68
	Basic P2F	699	16474	46661	1.09	2.69
Full model	P2F	735	7379	49588	1.08	3.84
(1975263 points)	F2P	595	-	-	2.75	2.75
	Combined	742	-	-	1.12	5.83
	Seedless P2F	691	7620	49499	1.13	3.86
	Static P2F	610	21345	49628	1.53	3.03
	Basic P2F	605	21117	49706	1.52	3.04
	Vocab. tree (all features)	677	-	-	1.4	4.0
	Vocab. tree (points only)	652	-	-	1.3	4.0

For each dataset, we evaluated the performance of localization and pose estimation using a number of algorithms. These include our proposed method and several of its variants, as well as a vocabulary tree-based image retrieval approach [6]. For each experiment, we accept a pose estimate as successful if at least twelve inliers to a recovered pose are found (we also discuss localization accuracy below). As in [4], we did not find false positives at this inlier rate (though some cameras had high localization error due to poor conditioning). The results of our experiments are shown in Table 2 - 4. For matching strategies, "F2P" and "P2F" denote feature-to-point and point-to-feature,

Table 3. Results for Rome. The entire test set consists of 1000 relevant images and 800 irrelevant ones (from Dubronik).

		Images	NN queries by P2F		Time in seconds	
		registered	registered	rejected	registered	rejected
Compressed model	P2F	921	12963	46756	0.91	2.93
(144777 points)	F2P	796	-	-	1.72	1.72
	Combined	924	-	-	0.87	4.67
	Seedless P2F	888	13841	46779	0.96	2.93
	Static P2F	805	21490	46966	1.35	2.87
	Basic P2F	808	21300	47150	1.35	2.88
Full model	P2F	863	11253	49500	1.57	4.34
(4067119 points)	F2P	788	-	-	2.91	2.91
	Combined	902	-	-	1.67	7.20
	Seedless P2F	769	10287	49635	1.52	4.33
	Static P2F	682	23548	49825	1.77	3.34
	Basic P2F	681	23409	49887	1.78	3.34
	Vocab. tree (all features)	831	-	-	1.2	4.0
	Vocab. tree (points only)	815	-	-	1.2	4.0

respectively, as described in Section 4. In "Combined", we use P2F first and, if pose estimation fails, rerun with F2P. The other three variants are simply stripped-down versions of P2F (Algorithm 1), with no seed points ("seedless"), with no dynamic prioritization ("static"), or with neither ("basic"). These are included to show how much performance is gained through each enhancement.

For Dubrovnik and Rome, the results for the vocabulary tree approach are obtained using our own implementation, using a tree of depth five and branching factor ten (i.e., with 100,000 leaf nodes). For each query image, we retrieve the top 10 images from the vocabulary tree, then perform detailed SIFT matching between the query and candidate image (similar to [4], but using actual images). We tested two variants, one in which all image features are used, and one using only features which correspond to points in the 3D model. When sufficient matches are found, we estimate the pose of the query camera given these matches.

All times for our implementation are based on running a single-threaded process on a 2.8GHz Xeon processor. For P2F matching, we show the average number of nearest-neighbor queries as well as running time for both images that are registered and those that fail to register. For F2P, however, these numbers are essentially the same for both cases since we exhaustively match image features to the model.

The results in the tables show that our point matching approach achieves significantly higher registration rates (without false positives) than the state of the art in 3D location recognition [4], as well as the vocabulary tree variants we tried. Among various matching strategies, the P2F approach (Algorithm 1) performs significantly better than its F2P counterpart. In some cases the results can be further improved by combining both together, at the expense of extra computation time. Although one might think the P2F would be slower than F2P (since there are generally more 3D points in the model than

Table 4. Results for Vienna

		Images registered	NN queries by P2F		Time in seconds	
			registered	rejected	registered	rejected
Compressed model	P2F	204	6245	32920	0.55	1.96
(200638 points)	F2P	145	-	-	2.04	2.04
	Combined	205	-	-	0.54	3.62
	Seedless P2F	182	6201	34360	0.54	2.07
	Static P2F	164	14393	39664	0.97	2.30
	Basic P2F	166	14274	40056	0.94	2.33
Full model	P2F	190	4289	41530	0.63	2.85
(1132745 points)	F2P	136	-	-	2.78	2.78
	Combined	196	-	-	0.71	5.32
	Seedless P2F	160	4034	44263	0.59	3.00
	Static P2F	162	16164	45388	1.11	2.72
	Basic P2F	161	16134	45490	1.11	2.67
Vocab. tree (from [4])		164	-	-	≤ 0.27 (GPU)	

features per image), this turns not to be the case. The experiments show that utilizing the extra information associated with the points makes P2F both faster and more accurate than F2P. The P2F variants that lack either dynamic priorities or seeding points, or both, perform much worse than the full algorithm, which illustrates the importance of these enhancements. Moreover, the compressed models generally perform at least as well as, if not better than, the full models, while being on average an order of a magnitude smaller in size. Hence they are able to save storage space and computation time without sacrificing accuracy. For the vocabulary tree approach, the two variants we tested are comparable, though using all image features gave somewhat better performance than using just the features corresponding to 3D points in our tests.

One further property of the P2F method is that when it recognizes an image (i.e. is able to register it), it does so very quickly—much more quickly than in the case when the image is not recognized—since if a true match is found among the seed points, the algorithm generally only needs to consider a small subset of points. This resembles the ability of humans to quickly recognize a familiar place, while deciding that a place is unknown can take longer. Note that even in the case where the image is not recognized, our methods is still comparable in speed to the vocabulary tree based approach in terms of equivalent CPU time, although vocabulary tree methods can be made much faster by utilizing the GPU; [4] reports that a GPU implementation sped up their algorithm by a factor of 15-20. Our point matching approach is also amenable to GPU techniques.

5.1 Localization Accuracy

To evaluate localization accuracy we geo-registered the initial model for Dubrovnik so that each image receives a camera location in real-world coordinates, which we treat as noisy ground truth. The estimated camera location of each registered image is then compared with this ground truth, and the localization error is simply the distance

between the two locations. For our results, this error had a mean of 18.3m, a median of 9.3m, and quartiles of 7.5m and 13.4m. While 87 percent of the images have errors below the mean, a small number were rather far off (up to around 400m in the worst case). This is most likely due to errors in estimated focal lengths (most likely for both the test image and the model itself), to which location estimates are very sensitive.

6 Summary and Discussions

We have demonstrated a prioritized feature matching algorithm for location recognition that leverages the significant amount of information that can be estimated about scene features using image matching and SfM techniques on large, heterogeneous photo collections. In contrast to prior work, we use points, rather than images, as a matching primitive, based on the idea that even a small number of point matches can convey very useful information about location.

Our system is also able to utilize other cues as well. While we primarily consider the visibility of a point when evaluating its relevance, another important cue is its distinctiveness, i.e., how well it can predict a single location (a feature on a stop sign, for instance, would not be distinctive). While we did not observe problems due to repetitive features spread around a city model, one future direction would be to incorporate distinctiveness into our model (as in [15] and [1]).

Our system is designed for Internet photo collections. While these collections are useful as predictors of the distribution of query photos, they typically do not cover entire cities. Hence many possible viewpoints may not be recognized. It will be interesting to augment such collections with more uniformly sampled photos, such as those on Google Street View or Microsoft Bing Maps.

Finally, while we found that our system works well on city-scale models built from Internet photos, one question is how well it scales to the entire world. Are there features in the world that are stable and distinctive enough to predict a single location unambiguously? How many seed points do we need to ensure good coverage, at least of the popular areas around the globe? Answering such questions will reveal interesting information about the regularity (or lack thereof) of our world.

References

1. Schindler, G., Brown, M., Szeliski, R.: City-scale location recognition. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2007)
2. Hays, J., Efros, A.A.: im2gps: estimating geographic information from a single image. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2008)
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. Irschara, A., Zach, C., Frahm, J.M., Bischof, H.: From structure-from-motion point clouds to fast location recognition. In: CVPR (2009)
5. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in video s. In: Proc. Int. Conf. on Computer Vision, pp. 1470–1477 (2003)

6. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 2118–2125 (2006)
7. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: Proc. Int. Conf. on Computer Vision (2007)
8. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2008)
9. Perdoch, M., Chum, O., Matas, J.: Efficient representation of local geometry for large scale object retrieval. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2009)
10. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring image collections in 3d. In: SIGGRAPH (2006)
11. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building rome in a day. In: ICCV (2009)
12. Simon, I., Snavely, N., Seitz, S.M.: Scene summarization for online image collections. In: Proc. Int. Conf. on Computer Vision (2007)
13. Se, S., Lowe, D., Little, J.: Vision-based mobile robot localization and mapping using scale-invariant features. In: Proc. Int. Conf. on Robotics and Automation, pp. 2051–2058 (2001)
14. Zhang, W., Kosecka, J.: Image based localization in urban environments. In: International Symposium on 3D Data Processing, Visualization and Transmission (2006)
15. Li, F., Kosecka, J.: Probabilistic location recognition using reduced feature set. In: Proc. Int. Conf. on Robotics and Automation (2006)
16. Ni, K., Kannan, A., Criminisi, A., Winn, J.: Epitomic location recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence 31, 2158–2167 (2009)
17. Jojic, N., Frey, B.J., Kannan, A.: Epitomic analysis of appearance and shape. In: Proc. Int. Conf. on Computer Vision, pp. 34–41 (2003)
18. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. Int. J. of Computer Vision 42, 145–175 (2001)
19. 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: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2009)
20. Crandall, D., Backstrom, L., Huttenlocher, D., Kleinberg, J.: Mapping the world's photos. In: Proc. Int. World Wide Web Conf. (2009)
21. Li, Y., Crandall, D.J., Huttenlocher, D.P.: Landmark classification in large-scale image collections. In: Proc. Int. Conf. on Computer Vision (2009)
22. Kalogerakis, E., Vesselova, O., Hays, J., Efros, A.A., Hertzmann, A.: Image sequence geolocation with human travel priors. In: Proc. Int. Conf. on Computer Vision (2009)
23. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building rome in a day. In: Proc. Int. Conf. on Computer Vision (2009)
24. Arya, S., Mount, D.M.: Approximate nearest neighbor queries in fixed dimensions. In: ACM-SIAM Symposium on Discrete Algorithms (1993)

Author Index